



# **Research Plan Us Employee Health Benefits SaaS Platform**

*Research Plan for a U.S. Employee Health Benefits SaaS  
Platform*

June 12, 2025

# Research Plan for a U.S. Employee Health Benefits SaaS Platform

- **\*Introduction:\*\*** This report presents a comprehensive research plan for building a Software-as-a-Service (SaaS) platform that enables U.S. employees to enroll or unenroll in employer-sponsored health plans, view detailed plan information, and search for in-network healthcare providers. The platform is envisioned to support small and medium businesses (SMBs), mid-market companies, and enterprise clients. We cover an overview of the U.S. health benefits landscape and stakeholders, technical architecture considerations for a scalable multi-tenant SaaS, compliance requirements (HIPAA, ACA, COBRA, etc.), integration strategies with insurance carriers (EDI 834) and HR/payroll systems, design requirements for plan comparison and provider search features, relevant vendor APIs for provider data, a competitive landscape analysis, key differentiators for market entry, and a proposed MVP scope with phased development recommendations. Actionable suggestions are included throughout to guide the platform's successful design and launch.

## U.S. Health Benefits Landscape & Key Stakeholders

- **\*Scale of Employer-Sponsored Insurance (ESI):** **Employer-sponsored health benefits are the primary source of coverage for Americans under 65. As of 2023, roughly 153 million non-elderly people in the U.S. receive health insurance through an employer. Employers (especially large ones) act as the** plan sponsors\*\*, offering group health plans to their workforce, often sharing premium costs with employees. This entrenched role of employers means any benefits platform must integrate smoothly into the

- employer-employee relationship.
- \*Key Stakeholders in ESI:\*\*
- **Employers (Plan Sponsors):** Employers select health plan options (e.g. medical, dental, vision plans) and manage benefit programs for their employees. They are responsible for plan design decisions, premium contributions, and ensuring compliance with relevant regulations. Most employers rely on external expertise – in fact, **79% of employers use brokers or benefits advisors to help identify and select benefit plans** for their employees. Employers, especially HR and benefits managers, will be primary users of the platform’s administrative functions. They need tools to manage enrollments, terminations, life event changes, and compliance reporting.
- **Employees (Plan Members):** Employees (and their dependents) are the end-users enrolling in plans. They value a smooth, informative enrollment experience that helps them understand options and make cost-effective decisions. Currently many employees are **dissatisfied with the complexity of getting coverage, understanding benefits, and finding care** – a recent survey shows they are “deeply unsatisfied with their experiences” in these areas. This underscores the need for a user-friendly platform that improves the member experience.
- **Insurance Carriers (Insurers/Payers):** These are the health insurance companies (e.g. UnitedHealthcare, Aetna, Blue Cross Blue Shield, Kaiser, etc.) that underwrite and administer health plans. Carriers maintain provider networks, process claims, and interface with benefits platforms to receive enrollment data. In the ecosystem, **health plans provide the insurance coverage and care management services, while employers and brokers play complementary roles**. Our platform will need to integrate with carriers to exchange enrollment information (e.g. via EDI 834 files or APIs) and possibly to retrieve plan details and provider network data. Carriers are also subject to data security and interoperability standards that the platform must accommodate.

- **Brokers and Benefits Consultants:** Brokers act as intermediaries advising employers on plan selection, cost management, and compliance. They often use benefits administration software to manage client enrollments. **Brokers' role is evolving to be more consultative, helping with compliance and digital tools in addition to plan shopping.** A successful platform should be broker-friendly (e.g. allowing broker access to set up plans for clients, or providing white-label capabilities) since brokers drive a lot of small and mid-market business. Notably, brokers appreciate technology integrations – **59% of employers said they would choose insurance carriers based on the carrier's ability to connect with their benefits technology platform**, which reflects how brokers and employers now expect seamless data connectivity between platforms and insurers.

- **Government & Regulators:** While not direct users, regulations from federal agencies (IRS, Department of Labor, HHS, state insurance departments) profoundly shape requirements. Laws like the Affordable Care Act (ACA), HIPAA, COBRA, and ERISA impose rules that the platform must support (discussed in a later section). For example, the IRS requires reports on coverage offers (ACA 1095-C forms) and the Department of Labor requires distribution of plan documents (ERISA). The platform will need to facilitate compliance for employers, effectively serving as a tool to meet regulatory obligations.

- **\*Market Segmentation:\*\*** Within the employer market, needs differ by size segment:

- **SMBs** (small businesses, e.g. under 100 employees) often lack in-house benefits expertise. They may rely on brokers or Professional Employer Organizations (PEOs) to handle benefits. SMBs need a very easy-to-use platform with minimal setup friction. They typically offer a few fully-insured plan options. Cost sensitivity is high, and they value simplicity and turnkey solutions (like Gusto or Ease – see Competitive Landscape).

- *Mid-market employers* (hundreds to a few thousand employees) have more complex needs: they might offer multiple plan choices, possibly across multiple states or carriers, and have to manage more compliance (ACA, etc.). They often work with brokers or benefits consultants. They value automation (to reduce manual HR work) and integration with their existing HR systems.
- *Enterprise* (large employers with thousands of employees) may be partially or fully self-insured, using third-party administrators (TPAs) or carrier ASO services. They often require custom integrations (HRIS, payroll, wellness vendors), robust security, and advanced features like evidence of insurability workflows, complex eligibility rules, and flexible reporting. They may conduct more sophisticated open enrollment campaigns and demand high configurability. Enterprise clients also require strict compliance support (e.g. ACA look-back measurements, COBRA administration, etc.) and may have global populations (though our initial scope is U.S.). A SaaS platform serving enterprise must be scalable and allow per-client customization (or even private instances in some cases).
- **\*Current Trends and Pain Points: The U.S. employer benefits landscape is complex and rapidly evolving. Healthcare costs continue to rise \~8-10% per year, pressuring employers to optimize benefits value. Employers are seeking ways to control costs while improving employee experience, leading to interest in technology solutions and innovative plan designs. However, many HR teams struggle with the growing complexity of benefits management and a “patchwork” of regulations, especially if they operate in multiple states. Compliance tasks (ACA reporting, COBRA notices, etc.) and administrative work (entering data in multiple systems) create risk and workload. These challenges are driving demand for modern, integrated benefits administration platforms. In fact, 70% of employers cite benefits digitization and systems integration as a top priority\*\* in benefits strategy. A key takeaway is that a new SaaS platform can add value by simplifying compliance, reducing manual work through integration, and**

- *empowering employees with better decision support.* The subsequent sections detail how to achieve this through technical and design choices.

## Technical Architecture Considerations for a Scalable SaaS

Building a scalable SaaS platform for benefits administration requires careful architectural planning. The system must securely support multiple client organizations (tenants) and thousands of users, with spikes in usage during open enrollment periods. Key considerations include multi-tenancy design, scalability, data isolation and security, and modularity for integration.

- **\*Multi-Tenant SaaS Architecture:** **To serve many employer clients efficiently, a multi-tenant architecture is ideal. In a multi-tenant model, a single instance of the software and database can serve multiple customer organizations, with appropriate data partitioning per tenant. This approach is cost-efficient and easier to maintain than one codebase per client. However, it demands robust security to** ensure each tenant's data remains isolated and protected from unauthorized access\*\*, and the system must be able to scale with increasing tenants and data volumes. We must decide on the degree of isolation per tenant:
  - **Pooled tenancy** – all tenants share the same application and database tables, with a tenant identifier to segregate data.
  - **Siloed tenancy** – each tenant has a separate instance or separate database/schema; this can simplify isolation at the cost of higher resource usage.
  - **Hybrid approach** – a mix where core services are multi-tenant (pooled) but certain components (e.g. a dedicated database schema or instance for very large clients) are isolated.

The architectural diagram below illustrates these models. The choice influences scalability and cost: pooled multi-tenancy maximizes cost efficiency and easier updates, while siloed provides stronger isolation and customization per client at higher cost. A mixed model can offer pooled resources for most clients and isolated resources for those that require it (for example, an enterprise client with special data policies could be on a dedicated database).

*Multi-tenant SaaS architectural models – pooled (shared resources for all tenants), siloed (separate resources per tenant), or a mixed approach. Each model has trade-offs in resource allocation, scalability, and cost optimization.*

For our platform, a **pooled multi-tenant model** with logical data isolation is likely suitable initially (to maximize scalability for many SMB and mid-market clients). We can implement tenant-aware logic in the application, using a tenant ID on all data records and scoping queries by tenant. Modern frameworks and cloud databases can enforce row-level security by tenant as well. It's critical that **each tenant's data is isolated and access-controlled**, both to prevent data leaks and to meet contractual privacy commitments (especially under HIPAA). As we onboard enterprise clients with stricter requirements, we can consider giving them isolated instances or databases (the mixed model) if needed.

- **\*Scalability & Performance:\*** The platform should be designed for horizontal scalability to handle periods of high load (e.g. an employer's entire workforce enrolling during open enrollment week). We should use cloud infrastructure (AWS, Azure, or GCP) with auto-scaling groups or Kubernetes orchestration so that additional application servers can spawn to handle traffic. A multi-tier architecture is recommended:
- **Web frontend:** A responsive web application (and possibly mobile app) that interacts with users. This could be built as a single-page app (React, Angular, etc.) for a snappy user experience, communicating via APIs.
- **Application layer (APIs/Microservices):** A set of stateless service endpoints (RESTful or GraphQL APIs) implementing business logic

- (enrollment rules, comparisons, etc.). Designing the system in a microservices style can help isolate domains (e.g., a microservice for enrollment transactions, another for provider search, another for compliance reporting) which can be scaled independently. Each microservice can incorporate **tenant context** in requests, applying tenant-specific rules or data filters. For example, a “Plan Management” service might be aware of an employer’s specific plan offerings and eligibility rules.
- **Database and Caching:** A robust relational database (with multi-tenant schema design) will store most data (employee info, elections, plans, etc.). We must ensure data partitioning by employer and implement encryption at rest for sensitive data (especially PHI). High-traffic data like plan documents or provider search results can benefit from a caching layer (e.g., Redis) to improve response times and reduce load on APIs.

Using a cloud provider’s managed services can offload some scalability concerns – e.g., AWS Aurora or Azure SQL hyperscale for the database to handle growing data, CDN for static content (plan brochures, etc.), and load balancers for traffic distribution. **Performance testing** should target open enrollment scenarios where thousands of employees might concurrently use the system; we’ll implement load testing to ensure the platform remains responsive (e.g., page loads under 2 seconds) under peak conditions.

- **\*Security by Design:\*\*** Because the platform will handle sensitive personal and health-related information, security is paramount. At the architecture level, we will:
  - Enforce strong **access controls and data isolation** by tenant. Users from one employer should never see another employer’s data. Role-based access control (RBAC) will ensure, for example, that only an employee sees their own enrollment info, and only authorized HR admins can see their organization’s roster.
  - Use **encryption** for data in transit (TLS for all network communication) and at rest (database encryption, encrypted file



- storage for documents). Protected Health Information (PHI) and Personally Identifiable Information (PII) must be stored and transmitted per HIPAA security rules (details in Compliance section).
- Implement audit logging for sensitive actions (viewing personal data, changing enrollments, etc.) to support security monitoring and compliance audits.
- Partition the system into network segments (using VPCs, subnets, security groups) so that database and internal services are not exposed directly to the internet.
- Regularly apply security patches and consider obtaining certifications or third-party audits (SOC 2, HITRUST) as we scale to enterprise clients.
- \*Enterprise Integration and Extensibility: **A scalable architecture should accommodate integration with external systems. We plan to** build an API layer for the platform itself\*\* – exposing secure APIs that allow, for instance, an HRIS to push a new hire to the benefits platform, or allow our platform to notify a payroll system of deduction changes. This API-first design not only enables integrations but also allows our own front-end to use the same APIs (ensuring consistency). Using widely accepted data formats and protocols (JSON, REST/HTTP or GraphQL, possibly SOAP for some legacy integrations) is important. In later phases, providing a library of webhooks (to notify client systems of certain events, like “Employee X completed enrollment”) and supporting SSO (e.g., SAML or OAuth for single sign-on with corporate directories) will be valuable for enterprise adoption.
- \*Cost Optimization and Tenant Resource Allocation: **We will leverage cloud scalability to optimize cost - e.g., multi-tenant resource pooling keeps costs lower for SMB clients. We’ll also monitor resource usage per tenant. If one large enterprise tenant heavily uses the system (e.g., running many reports), the architecture should prevent performance degradation for others (through rate limiting or workload isolation strategies). In a SaaS model,** efficient multi-tenancy is key to cost optimization\*\*, as noted in AWS’s SaaS best practices. As the number of tenants grows,

- we might implement sharding in the database by tenant groups to distribute load, and use caching aggressively for common data (like standard plan information).
- *\*Example Architecture Diagram (High-Level):\**
- *Clients:* Web Browsers/Mobile Apps of employees and admins -> (via Internet with TLS) -> *Load Balancer* -> *Web/App Servers* (stateless, containerized, auto-scaling) -> *Microservices & APIs* (enrollment service, provider search service, compliance service, etc.) -> *Database* (multi-tenant, encrypted) and *Cache*; plus integration connectors (for EDI file generation, API calls to carriers, HRIS integration points).
- *Supporting components:* A background job queue for processing batch jobs (e.g. generating EDI 834 files or large reports asynchronously), and a file storage service for documents (plan PDFs, SBCs, etc.).
- *Security components:* Identity provider integration for SSO, monitoring and logging pipeline, and backup/DR systems.

By thoughtfully designing the technical architecture in this manner, we ensure the platform can **scale to thousands of employers and users, maintain performance during critical periods, and remain flexible for future enhancements**. The next sections discuss in more detail how we will handle compliance, integrations, and specific feature designs within this architecture.

## Compliance Requirements (HIPAA, ACA, COBRA, etc.)

Compliance is a crucial aspect of any benefits administration platform. The system must help employers **meet legal requirements** related to offering health benefits and handling sensitive data. Key laws and regulations include:

- **HIPAA (Health Insurance Portability and Accountability Act):**  
This law mandates the privacy and security of Protected Health

- **Information (PHI).** In our context, PHI includes any personally identifiable health information – for example, an employee’s medical plan enrollment, SSN, or health status if collected. The platform must implement **HIPAA Security Rule safeguards** (administrative, physical, technical) to protect data. **HIPAA sets standards for privacy and security of health information.** Practically, this means using encryption, access controls, audit logs, and breach notification processes. If our platform is exchanging data with carriers or handling enrollment on behalf of a health plan, we are likely a *Business Associate* under HIPAA and will need to sign Business Associate Agreements (BAAs) with clients or carriers, committing to HIPAA compliance. We will need to restrict access to PHI on a need-to-know basis and provide features like secure messaging when communicating any health-related info. **Data security is paramount** – mishandling PHI could lead to severe fines. Our development and operations must follow best practices (e.g., secure coding, vulnerability scanning, employee HIPAA training).

- **ACA (Affordable Care Act):** The ACA introduced employer mandates and reporting requirements. Applicable Large Employers (50+ full-time equivalent employees) must offer affordable, minimum essential coverage to full-time staff or face penalties. **ACA requires these large employers to offer coverage or pay penalties.** The platform should assist with ACA compliance by tracking employee eligibility and enrollment. Key needs:

- **Measurement Period Tracking:** For variable-hour or seasonal workers, ACA allows measuring hours over time to determine full-time status. Our system could automate tracking of hours (if integrated with payroll or if hours data is input) and flag when employees become eligible.

- **Affordability and Minimum Value checks:** Ensure the plans an employer offers meet ACA’s affordability threshold and coverage requirements. While plan design largely happens outside the platform, we can provide tools or data to check if the lowest-cost plan for an employee is below the income percentage threshold.

- **IRS Reporting (1094-C/1095-C):** The platform should collect and store the data needed for ACA reporting each year (who was offered coverage, for what months, if enrolled or waived). Ideally, it can generate the 1095-C forms for employees and 1094-C for the employer, or at least export the data in a format that can be fed to a reporting tool. Many benefits systems incorporate ACA reporting modules. Our software should track coverage offers and waivers meticulously for this purpose (e.g., capturing if an employee waived because they have other coverage, etc.).
- Provide alerts for key ACA compliance tasks – e.g. if an eligible employee hasn't been offered coverage by their 90th day (ACA waiting period limit), or if an IRS affordability safe harbor might be violated based on the contributions entered.
- The **ACA compliance burden is significant**, including tracking full-time status, managing look-back periods, distributing **1095-C forms**, and responding to any IRS notices. Our platform should aim to **automate ACA tracking and form generation**, reducing the manual work and risk for employers.
- **COBRA (Consolidated Omnibus Budget Reconciliation Act):** COBRA gives employees (and dependents) the right to continue their employer-sponsored health coverage for a period after certain qualifying events (like termination of employment or reduction in hours). **COBRA obligates employers to allow continuation of coverage after leaving employment, within strict notification timelines.** Compliance requirements:
  - **Qualifying Event Tracking:** The platform should identify when a COBRA-qualifying event occurs (e.g., an employee is terminated in HRIS or marked as leaving in our system) and trigger the COBRA process.
  - **Timely Notices:** Employers must send COBRA election notices to qualified beneficiaries usually within 14 days of being notified of the event. While many employers outsource COBRA administration to specialized vendors, our platform can at minimum generate the data needed (e.g., a report of terminations with addresses) or integrate with

- COBRA service providers. In future phases, we might build COBRA notification functionality directly, but MVP could simply flag COBRA events for the HR admin to handle or send a file to a vendor.
- **COBRA Enrollment:** COBRA enrollees are typically handled separately (since they pay full premium). Our platform should be able to maintain COBRA participants' coverage records (perhaps marking them as COBRA in the system) and include them in carrier enrollment transmissions. Alternatively, if a COBRA TPA is used, the system might just interface by sending initial event info.
- Ensure that dependent events (e.g., divorce, over-age dependent) are also captured if those are in scope.
- **ERISA (Employee Retirement Income Security Act):** ERISA governs employer-sponsored benefit plans (including health plans). It imposes fiduciary responsibilities and requires certain disclosures to plan participants. **ERISA requires employers to provide plan information (like Summary Plan Descriptions) and follow fiduciary standards.** For our platform:
  - We should have the ability to store and distribute plan documents (SPDs, SBCs). For instance, an HR admin can upload their plan's SPD PDF to our system, and employees can access it in their portal. The Affordable Care Act also specifically requires distributing a **Summary of Benefits and Coverage (SBC)** for each plan – we can host these or link to carrier-provided SBCs.
  - ERISA also mandates tracking enrollments accurately and handling any required plan notices. Our system can help by time-stamping enrollments and providing confirmation statements (useful if an enrollment decision is ever in dispute).
  - Maintaining an audit trail of when employees were offered coverage and what they chose helps demonstrate ERISA and ACA compliance if needed.
  - While the platform can't perform fiduciary duties, it can equip employers with information (e.g., reports on contributions,

- nondiscrimination testing data for certain plans like Section 125 cafeteria plans) to fulfill their obligations.
- **IRS Rules for Tax-Advantaged Accounts:** Many employers offer Health Savings Accounts (HSAs), Flexible Spending Accounts (FSAs), or similar. While these might be administered by third-party vendors, our platform should not hinder compliance with their rules:
  - For HSAs, ensure only those enrolled in a high-deductible health plan can elect an HSA (we can enforce that in the enrollment flow).
  - For FSAs, ensure the annual election doesn't exceed IRS limits (and perhaps provide warnings if an employee tries to elect over the limit).
  - The platform could integrate with FSA/HSA providers or at least export enrollment data to them.
- **State and Local Laws:** We must also stay aware of state-specific regulations. For example, some states have mini-COBRA laws for small employers, some require reporting of offers of coverage to state agencies, and there are emerging requirements for benefits (like state mandates on fertility coverage or commuter benefits). Initially, focusing on federal compliance is priority; as we scale, we'll incorporate more state-specific logic or configurable rules. We will design the compliance engine to be updatable as laws change (e.g., ACA affordability percentage changes annually – the system should allow updating that value without code changes).
- \*Compliance Best Practices in Platform Design:\*\*
- **Automation and Alerts:** The platform should automate as many compliance tasks as possible and provide alerts for those that need human action. For example, automated tracking of hours for ACA, auto-generation of 1095-C forms, and alerts like “COBRA notification due for John Doe who terminated on X date” can drastically reduce the risk of things “slipping through the cracks”. Selerix (a competitor) notes that benefits administration software can **automate ACA tracking, HIPAA protocols, COBRA timelines, and IRS form generation to reduce errors**. Our platform should strive to include similar automation

- in later phases, if not in MVP.
- **Data Security (HIPAA):** We will implement features to help employers maintain HIPAA compliance. This includes **user account security (strong passwords, multi-factor authentication)**, role-based access (so that, for instance, a manager can't see an employee's health info unless authorized), **automatic logoff** after inactivity, and **audit logs** of who accessed what data. We should also ensure that if any PHI is emailed (e.g., an enrollment confirmation might contain plan selections), it is done securely or minimal PHI is included. Ideally, provide secure portals for employees to view information rather than sending sensitive info via email.
- **Audit and Reporting:** Compliance often comes down to having proper records. The platform should offer reports or data exports to satisfy audits. For example, a report of all employees, their full-time status, offer of coverage, and enrollment status is useful for ACA compliance audits or responding to IRS 226-J letters. Keeping history of benefit elections and changes is also important (e.g., if an employee claims they enrolled but the system shows they waived, the records and timestamps should be available).
- **Staying Current:** Regulations change. We will establish a process (perhaps a compliance advisory team or regular review of IRS/DOL updates) to update the software as needed. For example, if new ACA reporting codes are introduced, or if a new federal mandate (like the recently introduced No Surprises Act or transparency rules) require changes (e.g., showing certain cost information), the platform must adapt. Having a flexible rules engine or configuration for compliance thresholds (like ACA full-time = 30 hours, which could conceivably change, or new state family leave benefit rules) will help the platform remain current without extensive re-development.

By embedding compliance features into the platform, we turn a pain point into a value proposition for customers. Employers will be attracted to a solution that **“makes compliance easy”**, and indeed many benefits software vendors highlight this. For instance, Ease advertises that online

benefits administration can **“avoid penalties and fines”** by simplifying ACA and ERISA compliance. Our goal is to deliver that peace of mind through intelligent software design.

## Integration with Insurance Carriers and HR Systems

A major success factor for a benefits platform is how well it **integrates** with external systems – primarily insurance carriers (to transmit enrollment data) and secondarily HRIS/payroll systems (to exchange employee and payroll deduction data). Streamlined integration reduces manual work, prevents errors, and creates a seamless experience.

### Carrier Integration via EDI 834 (and APIs)

- **\*EDI 834 Standard: The primary method to integrate with health insurance carriers today is the HIPAA 834 EDI file, which is the standardized enrollment and maintenance transaction format. An 834 file is essentially a batch list of enrollment changes (adds, drops, updates) that the employer or its benefits system sends to the insurance carrier, typically via a secure FTP on a scheduled basis (e.g., nightly or weekly). The** EDI 834 transaction set is specified by HIPAA 5010 for electronic exchange of member enrollment information, including details of benefits, plan selections, and member demographics\*\*. It is used to communicate all enrollment events:
  - New enrollments (e.g., a new hire joining a plan)
  - Changes in enrollment (e.g., adding a dependent, or changing plan during open enrollment)
  - Reinstatements (re-activating coverage if someone was mistakenly terminated or returns to work)



- Terminations of enrollment (disenrolling members who left or dropped coverage)

These correspond to the key functions our platform must handle. Each 834 file contains records for subscribers and dependents, their chosen plan, coverage level, effective dates, etc. After sending an 834, the carrier returns a **999 acknowledgment** confirming the file receipt and whether the data passed format validations. Our system will need to process these acknowledgments and flag any errors for correction.

- \*Implementation: **During implementation, for each insurance carrier a client uses, we'll need to set up an EDI feed. This involves mapping our data fields to the carrier's required 834 layout (often carriers have slightly different companion guides specifying how they want the file). Many carriers still use SFTP file transfers for EDI. We should incorporate an EDI module or use a clearinghouse service to generate 834 files from our database. There are third-party solutions that manage EDI translation and transmission, or we could build it in-house given it's a core capability. Initially, focusing on generating accurate 834 files is critical for the platform's viability - benefits software must reliably send enrollment updates to carriers\*\* to be useful.**

We will schedule automated 834 generation for events:

- Regular full files or changes-only files during open enrollment (to send all employees' new year elections).
- Ongoing maintenance files for mid-year changes (new hires, terms, life events). These might be queued daily or weekly depending on client preference and carrier ability.
- It's advisable to start with a **daily changes file** approach - accumulate any changes in a day and send one file per carrier at day's end.

Our integration engine should also handle:

- **Carrier-specific custom fields:** e.g., some carriers require site/location codes or division identifiers on the enrollment records.
- **Error handling:** If a carrier's 999 or subsequent 834 error report indicates an issue (like someone's SSN already on file, or an invalid address), the platform should alert an admin to resolve it. Eventually we could build a dashboard for EDI status.
- **Reconciliation:** Periodically, the platform could import carrier eligibility files (some carriers send back a weekly roster) to ensure our records and the carrier's records match.
- **\*Modern API Integration: While EDI is prevalent, there is a trend toward real-time API integration with carriers. Some carriers and intermediaries are introducing APIs that allow adding or updating enrollments instantaneously. For example, startups like Noyo provide an API layer that connects to multiple carriers behind the scenes, enabling transactions without flat files. Using such solutions could be a differentiator for us:** "one single integration can enable seamless data connections with virtually any carrier"\*\*, replacing the need to build dozens of individual EDI feeds. In the long term, adopting API-based carrier connectivity can greatly improve speed and reduce errors (no batch delays). Noyo's data shows customers can activate carriers in days and cut operational costs by 60-70% versus traditional EDI methods.

For MVP, we will likely implement standard EDI to cover the necessary ground (as carriers all accept 834s). As we refine the platform, we can partner with API aggregation services (e.g., Noyo, Ideon/Vericred for enrollment, or directly with carriers that offer APIs) to offer **real-time or near-real-time connectivity**. This would allow, for instance, an employee's enrollment choice to be pushed to the carrier and confirmed within seconds, rather than waiting for a batch file. Such capability could become a key differentiator.

It's worth noting that **integration capability is a top criterion for employers selecting carriers** – nearly half of employers said they would switch carriers if integration was lacking. Thus, by providing excellent

carrier connectivity (whether via flawless EDI or modern APIs), our platform adds value to both employers and carriers (fewer errors and manual processes).

- **\*Carrier Data Feeds (Plan info, Rates, etc.): Apart from sending data to carriers, we also need to** receive data from carriers\*\* for certain functions:
- **Initial plan configuration:** The platform will need details of each insurance plan offered (plan names, network, deductibles, copays, rates tiers by family status, etc.). Often, brokers or employers input this manually during setup. However, services like Vericred (Ideon) offer **health plan design and rate data** via API for all major carriers. We could leverage such services to populate plan options (especially for small group plans) automatically, saving setup time.
- **Provider networks:** For the provider search feature, we may get provider-directory data from carriers or via an aggregator (detailed in the next section).
- **Ongoing updates:** If carriers update rates or plan offerings annually, an integration or data import (like uploading a spreadsheet of new rates) can help refresh the system for open enrollment.

In summary, robust carrier integration is non-negotiable. Initially, this means building a solid EDI 834 pipeline (with monitoring and error-handling). Over time, migrating to or supplementing with APIs will improve the experience. The integration layer must be designed with **scalability (to add many carrier connections)** and **configurability (different rules per carrier)** in mind. We might develop a mapping interface so that adding a new carrier feed is a configuration task, not a code change. This will accelerate onboarding new clients who use various insurance carriers.

## HRIS and Payroll Integration (Optional but Valuable)

While not strictly required for a functional MVP, integrating with HR and payroll systems can significantly enhance the platform's value, especially

for larger customers:

- **HRIS (Human Resource Information System) Integration:** The HRIS is typically the system of record for employee data (hire dates, job info, personal details). By integrating our benefits platform with HRIS, we can automatically sync employee demographic data and employment status changes. For example, when a new employee is added in the HRIS, an API or file integration can **automatically create that employee's profile in the benefits platform** so they can enroll without duplicate data entry. Similarly, terminations entered in HRIS can trigger the platform to terminate benefits (and possibly kick off COBRA). This eliminates HR having to enter the same info twice. According to industry guidance, modern HRIS integration often uses either **flat-file transfers (EDI feeds) or real-time APIs/webhooks**:

- *File-based (EDI/CSV) integration:* The HRIS can produce a file (daily or weekly) of employee changes which our system imports. This is reliable but not real-time.
- *API integration:* Many modern HR systems (Workday, BambooHR, Namely, etc.) have APIs to query employee data or receive event notifications. Using APIs or middleware (like an iPaaS platform or a service like Merge or Finch which specialize in HRIS integrations) enables near real-time updates. **APIs and webhooks allow real-time, flexible data sharing**, which is ideal for enterprise needs.

We should prioritize building API connectors for the most common HR platforms our target clients use. However, given the variety of HRIS in the market, providing a flexible flat-file import as a fallback is wise for MVP (e.g., the HR admin can upload an Excel of employee census to load initial data).

- **Payroll Integration:** Payroll integration is important for handling **premium deductions**. Once an employee enrolls in benefits, the employer needs to deduct the appropriate premium from their paycheck. If our platform integrates with payroll, it can automatically update deduction codes/amounts for each employee's selected benefits, sparing HR from manually entering deductions and reducing

- errors. For example, Gusto (which is both a payroll and benefits system) automatically syncs the benefits premiums with payroll so that deductions are always up to date. Our platform could achieve this by:
- Generating a report or file of deduction changes each pay period that can be uploaded to the payroll system.
- Building direct API integration with popular payroll providers (e.g., ADP, Paychex, QuickBooks Payroll) to push changes. Companies like Paycor note that integrating benefits with payroll **avoids errors and redundancies by syncing data**.

The advantage is significant: **automating salary and deduction updates** reduces administrative burden, ensures accuracy, and improves compliance in tax reporting. For instance, incorrect deductions could lead to incorrect taxable wage calculations, so integration helps get it right.

- **Use Cases and Priority:** For SMB clients, sometimes using an all-in-one (like Gusto or Rippling) is popular because of these integrations. Since our platform is standalone, providing integration options makes it competitive. We might prioritize integration with:
- Common SMB HR/payroll systems: Gusto (though Gusto is itself a competitor, some clients might use our platform for benefits if they don't use Gusto's), ADP, Paychex, Paylocity, BambooHR, Namely.
- Common mid-market/enterprise systems: Workday, Oracle PeopleSoft, SAP SuccessFactors, UKG (UltiPro), etc. These often have established integration approaches (Workday, for example, often shares benefits data via its PEGI or Integration Center; others might do SFTP files).

We likely cannot build all at once, so a phased approach is needed. MVP could launch with a **simple CSV export** that an admin can download from our system with all deduction info and then import into their payroll system – a manual but workable solution. In later phases, implement direct connectors for the most requested systems.

- **Standards:** There are some data standards in HR integration (like the HR Open Standards, formerly HR-XML, for data exchanges). If feasible, adopting a standard format for exchanging data (XML or JSON schemas that some HR systems support) can simplify mappings. For example, ADP has APIs, Workday can produce 834-like files for benefits (some HRIS can output an 834 file as if they were the source). In fact, SAP SuccessFactors has a built-in **834 file output template** that can be used to feed our platform or carriers. We can leverage these where available.
- **\*Integration Best Practices:** **We will ensure that integration points are** secure and reliable\*\*. Data transfers should be encrypted (SFTP or API with HTTPS). We'll implement error handling (e.g., if an API call to update payroll fails, flag it to admin). During implementation with a client, we'll do an initial data sync (upload current census, etc.) and then turn on ongoing sync.

One caution: not all clients will choose to integrate all systems (some may prefer to handle certain things manually). So our platform should function even without integrations – i.e., HR can key in a new hire manually if needed. Integration is a value-add that we will strongly encourage due to efficiency gains.

- **\*Summary of Integration Benefits:\*\*** By connecting to carriers and HR systems, our platform becomes a central hub that ties everything together:
- **Less Manual Data Entry:** HR doesn't have to enter the same information in multiple places (HRIS, benefits system, carrier forms, payroll). This *streamlines operations and improves data consistency*.
- **Fewer Errors:** Integration means fewer transcription mistakes. Consistency between systems improves compliance (e.g., tax filings and benefit reports align).
- **Real-time Updates:** Especially with APIs, any change (hire, term, life event) can flow through immediately, ensuring timely enrollment changes (important for avoiding coverage gaps or over-deductions).

- **Better Employee Experience:** For example, an employee changes address in the HR portal; if integrated, that update reflects in their benefits enrollment and is sent to the carrier – so their insurance ID card is mailed to the correct address. Or an employee’s payroll deduction adjusts automatically when they switch plans, so their paycheck is accurate.

Given these advantages, integration capabilities will be **a selling point of the platform**. We will highlight that our system “plays nice” with others. Indeed, one guide suggests that when choosing benefits software, ensure it **“integrates seamlessly with your HRIS, payroll, and compliance platforms”**. We plan to meet that expectation by offering a range of integration solutions tailored to client needs and technical capabilities.

## **Design Requirements: Plan Comparison, Enrollment Flow, Provider Directory**

The user experience (UX) of the platform is critical to its adoption. The platform should simplify a traditionally complex process. Below are design requirements in three core areas: plan information & comparison, the enrollment workflow, and provider search functionality. We aim for a **clear, intuitive interface** that caters to employees (making decisions) and HR admins (configuring and monitoring enrollment).

- \*Plan Information and Comparison:\*\*

Employees often struggle to understand their benefits options, so the platform must present plan details in a **transparent, easily digestible way**. Key design features:

- **Plan Detail Pages:** Each health plan (or benefit option) should have a page or modal with comprehensive information: premium per pay period, deductible, out-of-pocket max, copays, coinsurance, covered services summary, and any notable features (e.g., HSA eligibility, in-network vs out-of-network coverage). We should also display required

- documents like the Summary of Benefits and Coverage (SBC) for those who want fine print. This page should use simple language and possibly icons or visuals to represent features (for example, a piggy bank icon next to HSA-eligible plans).
- **Side-by-Side Plan Comparison:** Employees benefit from comparing plans directly. The platform will offer a **plan comparison tool** allowing the user to select two or more plans and view their features side by side. This could be presented as a comparison table listing key attributes (premium, deductible, etc.). For instance, Oracle's PeopleSoft system has a "Fluid UI" plan comparison feature, and Workday's benefits interface emphasizes plan comparisons and cost details. We will emulate best practices from such systems. **Employees should be able to compare plan options side-by-side with clear, concise information** to decide which coverage is best for them. By showing differences (perhaps highlighting the cheapest premium, the lowest deductible, etc.), we facilitate informed decision-making.
- **Cost Calculation and Decision Support:** Beyond static comparisons, incorporate tools to help estimate total cost. For example, a **"plan cost calculator"** could let employees input expected healthcare usage (like number of doctor visits, anticipated prescriptions, etc.) and then estimate out-of-pocket costs under each plan. While an advanced feature, even a simple calculator that sums up annual premium + expected out-of-pocket can be helpful, especially when comparing a high-deductible plan vs a higher-premium low-deductible plan. Many employees focus only on premiums; our UX should gently educate them to consider worst-case out-of-pocket exposure too.
- **Personalized Recommendations (future phase):** As a stretch goal, the platform could suggest an optimal plan based on personal circumstances (if the user chooses to input data like preferred doctors, medications, or risk tolerance). For MVP, we may not include a full recommendation engine, but we can include help text like "Things to consider when choosing a plan" and highlight if, for example, one plan is HSA-eligible and the user could save taxes with an HSA.



From a design standpoint, **simplicity is key**. Use plain language (avoid insurance jargon or explain it via tooltips). For example, label “Primary Care Visit: \$20 copay” rather than burying that in a paragraph. Use **progressive disclosure**: basic info upfront, with the ability to click “more details” for those who want the nitty-gritty. Remember, many employees find this confusing; a McKinsey study noted employees want an improved experience in **“understanding benefits”** and are often unsatisfied, so a user-friendly comparison can set us apart. In effect, the platform should act as a digital benefits counselor, presenting options clearly and *perhaps injecting guidance through UI/UX*.

- **\*Enrollment Flow:\*\***

The enrollment process should be a guided step-by-step flow, minimizing confusion and ensuring required data is collected. Typical steps and design elements:

1. **Welcome and Instructions:** If it's an open enrollment period or a new hire enrollment, have a welcome screen explaining the deadline and steps. Outline how many steps there are (e.g., 5 steps: Profile -> Dependents -> Plan Selection -> Review -> Confirm). This manages user expectations.

2. **Profile Confirmation:** Ask the employee to confirm key personal info (address, DOB, etc.) and maybe any info needed for benefits (marital status, smoker status if relevant to rates). If pulling from HRIS, this is just a “review your info” step.

3. **Dependent Management:** If the employee can enroll dependents, provide a screen to add/edit dependents (spouse, children). Collect names, DOB, SSN, etc., as needed by carriers. Make it optional if they don't have any. Allow them to indicate which dependents they want to cover for each benefit (this can also be done at plan selection step by toggling coverage tier).

4. **Plan Selection:** This is the core step. It should list the categories of benefits (Medical, Dental, Vision, Life, etc.). Under each category, present the plan options available. The UI might be a tile for each plan with its

name, a few key highlights, and the cost to the employee. The employee selects one plan per category (or “Waive” if they decline that benefit). For medical, if multiple plans are offered, they choose one. If they waive, we may prompt a reason (some employers collect waive reasons like “Covered under spouse’s plan” – not mandatory, but nice to have for ACA reporting).

When a plan is selected, if dependents exist, the system should ask which dependents to cover under that plan (i.e., selecting coverage tier: employee-only, employee+spouse, etc. could be implicitly determined by which dependents are checked). This can be an interactive step – for example, the user checks boxes next to each dependent’s name to include them, and the UI updates the cost accordingly (premium changes for single vs family coverage).

We will incorporate **decision support cues**: for example, a note like “Your doctor Dr. Smith is in-network for this plan” if we have provider matching (see provider search below), or “This plan pairs with an HSA – contribute to save tax” as guidance. The selection page can have a “Compare these plans” link that brings up the side-by-side view if the user wants to review again before choosing.

The design should clearly display the **per-paycheck cost** to the employee for each option, as cost is a primary decision factor. It might also show the employer contribution, but that’s secondary for the employee’s choice (though transparency is good).

**5. Additional Benefits or Questions:** If the employer offers other benefits (like voluntary life insurance, disability, FSA, etc.), those would have similar selection steps. Some may have decision support (e.g., for life insurance, maybe a suggested coverage amount or showing how much costs by coverage level). Keep the flow moving but allow skipping if not interested (e.g., “I do not want to enroll in a Healthcare FSA”).

**6. Review and Confirmation:** Summarize the chosen benefits and covered dependents on one page. Show the total per pay period deduction and maybe per-year cost. Prompt the user to review and then **confirm enrollment**. We might include an electronic attestation like “I agree that

the above selections are correct and authorize payroll deductions” to serve as a digital signature. Once confirmed, present a confirmation screen or downloadable confirmation statement (and send an email confirmation if appropriate).

**7. Post-Enrollment:** The user should be able to return anytime to view their current enrollments. If during an open enrollment window, they might be allowed to change their elections up until the deadline (in which case, we’d show an “Enrollment complete – you can make changes until \[date]” message). Outside of open enrollment, changes would be locked except for qualified life events, which we should support via a separate flow (life event initiation triggers a special enrollment window – likely a later phase feature).

Throughout the enrollment flow, the UI must be **responsive (mobile-friendly)** since some employees may complete enrollment on their phone. The steps should be clear with back/next navigation and a progress indicator. We should aim to keep each screen from being overwhelming – better to have multiple small steps than one gigantic form.

- **\*Provider Search and Directory Access:\*\***

Having an integrated **provider search tool** is a key requirement. Many employees base plan decisions on whether their preferred doctors or hospitals are in-network. Thus, our platform should enable users to **search for healthcare providers and see which plans those providers participate in.**

Design and functional considerations for provider search:

- **Search Interface:** Provide a dedicated page or modal for provider search, accessible during plan selection (e.g., a button “Find a doctor”). The search should allow queries by provider name, specialty, facility, or other criteria (e.g., “Dr. John Smith” or “Cardiologists in ZIP 10001”). We should also allow filtering by location (ZIP code or city) and distance, since users often want a doctor near them.
- **Network/Plan Filtering:** The tool should either ask the user to specify which plan’s network to search, *or* more powerfully, allow

- searching across all available plans' networks simultaneously. For example, a user could search "Dr. Jane Doe" and the tool returns: Dr. Doe is *in-network* for Plan A and Plan B, but *out-of-network* for Plan C. This information is **extremely helpful** for plan choice. Vericred's API, for instance, allows queries of which plans a given doctor accepts. Our implementation can leverage such capabilities. The Ideon blog highlighted that **integrating provider-network data simplifies finding in-network doctors and can even drive personalized plan recommendations**. So our platform should use provider data to guide employees: e.g., "If you want to keep seeing Dr. Doe, choose a plan that includes **Network X**."
- **Data Source:** To power this, we need up-to-date provider directory data. We have a few approaches:
  - **Carrier APIs/feeds:** With new regulations, many carriers have to expose provider directory info via **FHIR APIs** (following the CMS Interoperability Rule and the DaVinci PDEX PlanNet standard). For example, HealthSparq (a Kyruus company) offers a FHIR-compliant Provider Directory API that third parties can query. We could query each carrier's API for providers. However, dealing with multiple carriers' APIs and authentication might be complex.
  - **Aggregators:** Use a service like **Vericred/Ideon** or **HealthSparq** as a one-stop provider data source. **Ideon (Vericred) offers a Provider Network API** that aggregates provider data across carriers, allowing queries like "find providers by name/location and filter by insurance network" and "given a doctor, list the plans that doctor accepts". This can dramatically simplify our development, as we integrate with one API instead of dozens of carriers. We likely will use such a vendor API in our platform (even if it adds cost, it's a worthwhile trade-off for comprehensive data).
  - **Manual links fallback:** For MVP, if we can't fully integrate an API in time, at minimum we should provide direct links to each carrier's own provider lookup webpage (many carriers have "Find a doctor" tools on their sites). That's not ideal, but better than nothing. However, the goal

- is definitely an integrated search to keep users in one place.
- **User Flow for Provider Search:** A likely use case: an employee is on the fence between Plan A and Plan B. They click “Find Your Doctor” and search Dr. X. The result shows Dr. X is in network for Plan B but not Plan A – the platform could highlight this clearly (perhaps with a checkmark next to Plan B). This information might swing their decision. Alternatively, a user might search for a new provider (e.g., find a primary care physician accepting Plan A within 5 miles). So the search results should display provider details: name, specialty, address, contact, and importantly, which plans (of those the employer offers) include that provider in-network. If a provider is not in any offered plan, we should say so (and maybe suggest searching a broader radius or confirm spelling).
- **Quality and Cost Data (future):** Some advanced navigation platforms (like HealthJoy, Castlight) also integrate provider quality ratings or cost-efficiency scores to guide users to high-value providers. In future, we could integrate such data (for example, via Medicare ratings or third-party data) to not only show who is in-network, but perhaps highlight “preferred” providers or cost estimations. For MVP, this is likely out of scope, but keeping the design extensible to add provider rating info would be wise.
- **Integration in Enrollment Flow:** We should embed prompts in the enrollment UI like, “**Check if your doctors are covered**” with a link to provider search. Also, when viewing plan details, listing a sample of major provider networks or hospitals covered by the plan can give a sense (e.g., listing the plan’s network name and maybe a few major hospital systems in-network). Ultimately, employees feel more confident enrolling if they know their doctors or preferred facilities are accessible.
- **Mobile Consideration:** Provider search might be used on the go. Ensure the map or list of search results is mobile-friendly. Potentially integrate with a mapping API to show locations on a map if time permits.

- \*Example of Integrated Provider Search Benefit: **Word & Brown (a benefits general agent) built an integrated provider search in their quoting platform, allowing brokers to** instantly search and confirm which plans include the medical providers their clients want. **They highlight that it eliminates guesswork and the need to separately search each carrier's site. Our platform's provider search will similarly aggregate all relevant data "in one place" - a strong differentiator against older systems where an employee might have to manually cross-check provider directories.** Offering all medical carriers' data in one integrated search ensures employees have the latest information without tedious external searches.\*\*

In summary, the design goal is to make the enrollment experience as **educational and stress-free as possible:**

- The platform should **simplify comparison** of complex plan options with visual aids and clear language.
- It should **guide the user step-by-step** through enrollment, validating inputs and providing helpful tips (for instance, if someone skips dental, maybe confirm "Are you sure you want to waive dental coverage?" just to prevent accidents).
- And it should **empower the user with tools like provider search** so they can make an informed choice about which plan will work best for them and their family.

A successful UX here will not only delight users but also reduce the burden on HR (fewer basic questions if the system is self-explanatory). We will conduct usability testing with sample users (perhaps employees from a pilot client or friends/family) to ensure the design is intuitive. Short, jargon-free content, logical flow, and robust help (FAQs, possibly live chat support integration for questions) are all design considerations aligned with creating a modern benefits platform.

# Provider Search Data Sources and Vendor APIs

Implementing a robust provider search feature requires sourcing comprehensive and up-to-date provider network data. Rather than reinventing the wheel, we can leverage external **vendor APIs and data sources** that specialize in health plan and provider information. The leading options mentioned include **HealthSparq, Vericred (Ideon)**, or direct carrier feeds. Here we evaluate these and how to incorporate them:

- **Vericred / Ideon API:** Vericred (now part of Ideon) is a well-known provider of health insurance data APIs. Their platform offers **access to essential health plan data, provider directories, and compliance data via API**. Specifically, Vericred's API features:
  - *Provider Directory Data:* A database of healthcare providers (doctors, hospitals, etc.) along with the insurance networks and plans they accept. Vericred maps provider-network affiliations at the plan level nationwide.
  - *Plan Search and Quote Data:* They have endpoints for searching health plans by zip code, getting plan details, rates, etc., which could help in building our plan library.
  - *Integration:* The API is designed to integrate into benefit software, insurance marketplaces, etc., with **friendly documentation and the ability to embed in our workflows**.
  - *Notable Users:* Companies like Oscar Health and major carriers use Vericred, indicating its reliability and breadth.

For our needs, the **Provider Directory endpoints** are most relevant. For example, we can use an endpoint to query providers by name or specialty and filter by insurance plan or network. Or given a specific plan ID, retrieve all providers in that network (though that could be thousands; usually we'd filter by location too). **Vericred's provider data includes provider details (name, specialty, contact) and network affiliations**, which covers everything needed for our search results. By

integrating this API, our platform can, with a single query, search across **all carriers' networks from a single source**. Ideon (Vericred) essentially aggregates data from many carriers, saving us from handling each feed.

Additionally, Vericred offers compliance-related data (like ACA metal tier info, etc.) and even an enrollment API product (IdeonEnroll) that might handle transmitting enrollments. We could explore using IdeonEnroll as an alternative to building our own 834 connections in the future. It can connect to carriers via API as a middleman.

- **\*Cost\*\***: Vericred is a paid service, typically charging per API call or on a subscription basis depending on number of lives. We'd factor this into our pricing model (perhaps an added cost per employee or per client for provider search capability). Given the value (time saved and richness of data), it's likely worth it.
- **HealthSparq / Kyruus**: HealthSparq provides provider search and cost transparency tools, often used by health plans themselves. They have developed **FHIR-compliant Provider Directory APIs** (per CMS requirements) that expose provider data in a standard format. Some health plans contract HealthSparq to manage their directories, which means if we query a plan's FHIR API, it might actually be hosted by HealthSparq. Using HealthSparq's API directly might require being a partner; however, since these are meant to be public (for interoperability) we could theoretically query each carrier's FHIR endpoint. But we'd have to do it carrier by carrier, handling authentication and data standardization.

HealthSparq/Kyruus also might offer a unified API product for third parties – though their focus is typically health plan clients, not software vendors. Still, HealthSparq's solution emphasizes accurate, validated provider data and even enhanced profiles (they have something called Validated Provider Profiles to meet regulatory requirements). If Vericred were not used, HealthSparq/Kyruus could be a plan B, but likely Vericred/Ideon is more directly geared for benefit software integration.



- **Carrier-Direct Feeds:** In absence of an aggregator, we could obtain provider data directly from each insurance carrier. Some carriers might provide periodic full directories (e.g., monthly data dumps of their network), possibly via their broker or client portals. However, merging and indexing all these would be a huge data engineering effort. Also, data changes frequently (providers join or leave networks, addresses change, etc.), so it's better to use a service that stays on top of updates. If we had a very large client base and wanted to avoid vendor fees, we might consider building an internal provider directory service – but as a new entrant, it's more efficient to rely on existing APIs.

It's worth noting that as of mid-2020s, under the 21st Century Cures Act and related rules, insurers are mandated to have provider directory APIs. We could leverage that by writing to the FHIR PlanNet specification to pull data. But again, this is complex to do for dozens of carriers and plan variations. So, using an aggregator like Ideon simplifies compliance with that rule and ensures our data is current.

- **Other Data Vendors:**

- *Quest Analytics (BetterDoctor):* Quest owns BetterDoctor, another provider data aggregator historically used for directory validation. Possibly not as accessible via API to us unless we contract it.
- *CLIQ or Ribbon Health:* There are startups like Ribbon Health offering provider data APIs with additional info (like quality metrics). Ribbon Health or similar could be integrated if we want to layer in provider cost/quality info in the future.
- *Government Sources:* For some providers, basic info is available via NPI registry etc., but not network affiliations. So not enough.
- \*Provider Search Implementation Plan:\*\*

For MVP, we plan to integrate the **Vericred (Ideon) API for provider search**. Steps:

1. *Obtain API access:* Register with Ideon for API keys and documentation. Decide on which endpoints to use (likely their Provider Search and

possibly Provider Lookup by NPI).

2. *Data mapping*: Ensure we have reference data aligning our plan offerings to Vericred's identifiers (Vericred assigns its own IDs for insurance plans). Typically, you pass a carrier ID or plan ID in the API query. We might need to map our internal representation of "BlueCross Gold PPO" to Vericred's network ID for that plan. Vericred does provide ways to search plans by metal level, area, etc., which could help populate our plan info too.

3. *UI Integration*: Build the front-end components for search input and results. The front-end calls our backend (to hide API keys) or directly calls the API if allowed (likely via our backend for security). We parse the JSON results and display providers in a user-friendly list (name, distance, accepting new patients indicator, etc.), with icons or labels for the plans they accept that match the employer's offerings.

4. *Performance*: Provider searches could potentially return many results. We'll implement pagination or "load more" for results beyond a certain count. We'll also want to cache frequent queries during an active enrollment season (if many employees search the same few popular doctors, caching their info for a short period can reduce API calls).

5. *Accuracy and Update Frequency*: We rely on the vendor to maintain accuracy. According to Ideon, using a single source like them ensures up-to-date info is readily available. They claim to deliver data via standard API which removes the need for us to aggregate from multiple sources.

- **\*Backup Plan:** If there are any hiccups or if a client's specific carrier isn't well-covered by the API, we will always offer links to carrier directories as a fallback (e.g., "Search on UnitedHealthcare's site" link). But this would be outside our ideal UX and used only if needed.
- **\*APIs for Other Features:**

While provider search is a highlight, vendor APIs can help with other functionalities:

- **Plan data APIs:** Ideon/Vericred has plan design and rate data for ACA market plans which could help populate plan info for small group plans. For example, if an employer in NY with 50 employees wants to offer a certain Empire Blue plan, and we have the plan ID, we can fetch details (deductible, etc.) rather than manually input. This reduces configuration time. Over time, building a library of plan templates from such data will streamline onboarding new clients.
- **Network Matching API:** Vericred also has an endpoint to **see which plans a specific provider accepts**. That's the inverse of typical search and could be used if, say, an employee enters their doctor up front and we then highlight the plans that include that doctor (even before they search explicitly in the UI).
- **Enrollment APIs:** Some vendors (like IdeonEnroll or Noyo) can handle the actual delivery of enrollment data to carriers via API. If we partner with one, our platform could call an API with an enrollment change and the service handles sending to the carrier (replacing the need for us to manage EDI files). This is an intriguing option to reduce our heavy lifting on the back-end integration. For MVP we assumed building 834, but we will evaluate such partnerships to accelerate carrier connectivity.
- \*Summary of Vendor Options:\*\*
- **Ideon/Vericred:** Comprehensive solution for both plan and provider data, well-suited for our needs (planned choice for provider search).
- **HealthSparq/Kyruus:** Another robust provider data source, aligned with FHIR standards – could be used especially if a client specifically requires using the carrier's official API for compliance.
- **Other APIs (e.g., Google Maps API):** We might use Google Maps API to geocode addresses or show map in provider search results.

Using these external data sources allows us to focus on the user-facing functionality rather than building massive provider databases. It aligns with the strategy of **“configuring instead of coding”** where possible, to speed up development. By integrating best-in-class data services, our

platform can offer features that rival much larger competitors from day one, which is important for market entry.

## Competitive Landscape Overview

The HR/benefits technology space has several established players and emerging startups. Understanding their offerings will help us position our platform and identify gaps to exploit. Below is an overview of relevant competitors, including those mentioned (HealthJoy, Gusto, Ease, Nava) and others in the ecosystem:

- **HealthJoy:** HealthJoy is a **digital benefits navigation platform** aimed at improving employees' utilization of healthcare and benefits. It provides a **mobile app with an AI-powered virtual assistant ("Joy") and a live concierge team**. HealthJoy aggregates all of an employer's benefit resources into one app for employees, making it easy to access telemedicine, EAPs, insurance info, and more. A key feature is its healthcare concierge that can help employees find in-network providers, explain benefits, and even tackle billing questions. HealthJoy does not replace a benefits administration system for enrollment; instead, it layers on after enrollment to drive better usage and understanding year-round. **HealthJoy focuses on a personalized, "Netflix-like" user experience for benefits, with ongoing engagement and cost transparency tools, guided by their concierge team**. They target mid-market employers often as a value-add service. For our platform, HealthJoy represents a potential partner or complementary service (some employers might use our platform for enrollment and add HealthJoy for navigation). If positioned as a competitor, its strength is in user engagement and healthcare navigation rather than the enrollment process itself. Our provider search and decision support features could take inspiration from HealthJoy's approach to guiding employees (e.g., using chatbots or interactive guidance in future). HealthJoy has raised significant funding (Series D) and has about 900k users, showing strong market traction for the problem of benefits engagement.

- **Gusto:** Gusto is an **all-in-one payroll, HR, and benefits platform** primarily for small businesses (SMBs). It started as a payroll service and added benefits brokerage/administration for small employers, acting as the broker of record for health insurance in many states. Gusto's value proposition is that it **makes it extremely easy to offer and manage benefits for small teams**, tightly integrating with payroll. For example, through Gusto an employer can get quotes for medical, dental, vision plans, choose one, and then employees enroll online. **Setup is simple, and payroll deductions are automatically synced** without extra work. Gusto also offers other benefits like 401k, life insurance, etc., often via partners, managing them through one dashboard. As a competitor, Gusto is very strong in the sub-50 employee market who want a one-stop shop. Its limitations might be in serving larger clients with complex needs (Gusto's HR features are fairly basic and it might not handle complex plan structures). Our platform, not offering payroll, must integrate well with payroll to compete in SMB and emphasize ease-of-use to match Gusto's "it just works" reputation. Notably, Gusto monetizes benefits by earning broker commissions. We could consider a similar approach or support broker integration. **Gusto's seamless payroll-benefits integration and licensed advisor support are key selling points.** SMB clients might choose us over Gusto if they already have a payroll system they like (or need more flexibility in plan options).

- **Ease (Ease.com):** Ease is a **benefits administration software focused on insurance brokers and SMBs**. It is often provided to small employers through their insurance broker to facilitate online enrollment and benefit management. **Ease makes it simple for brokers and small businesses (2-250 employees) to set up and manage benefits enrollment online.** It digitizes what used to be paper forms. Ease's strategy is to partner with brokers – over 2,300 agencies use Ease, giving it reach into tens of thousands of small employers. Notable features: employee self-service enrollment, a library of carrier forms and plan information to speed setup, basic HR functions (onboarding, org charts), and some compliance tools (ACA reporting,

- etc.). Ease is rated highly in the SMB segment. One reason for its success is that it's free or low-cost to employers; brokers often cover the cost to streamline their own operations. As a competitor, Ease has first-mover advantage in broker distribution. To differentiate, our platform could offer a more modern UX (Ease is good but has room to improve UI), more advanced features typically not offered to SMB (like integrated provider search or AI support), or tackle mid-size companies that have outgrown Ease. We might also consider integrating with brokers similarly. **Ease's strength is in being purpose-built for broker workflows and small group benefits, providing an easy online enrollment system.** Any new entrant in that space must either complement broker workflows or offer a new channel (e.g., direct-to-employer with more services).

- **Nava Benefits:** Nava is a **tech-enabled benefits brokerage** targeting mid-market employers. Their model combines traditional brokerage services (helping design benefit strategies and negotiate with carriers) with a proprietary technology platform for both HR and employees. **Nava's platform ("Nava HQ") provides an intuitive member experience with 24/7 support, and an AI-powered assistant that can answer employees' benefits questions in real-time.** They emphasize features like:

- Real-time renewal modeling for HR (to visualize how changing plans or contributions will affect budgets).
- An employee app/portal with an AI chatbot that can, for instance, tell an employee how much of their deductible they've used or guide them to a nearby in-network clinic.
- Compliance and document storage tools.
- A "benefits HQ" that centralizes all benefits management for HR.

Nava's differentiator is combining high-touch service (each client has a Nava benefits consultant team) with modern software. They market themselves as **"the benefits team you trust, supercharged with technology"**, indicating a hybrid of service and SaaS. Nava typically is

the broker of record for their clients, earning commissions, and uses their software to deliver better outcomes (they claim to focus on cost-saving strategies, etc.). They're newer (founded ~2019) but have grown by pitching a fresh approach to frustrated HR teams.

For us, Nava is both a competitor (if an employer chooses Nava as broker, they'd use Nava's platform, not ours) and a validation that better software + expert support is a winning formula in benefits. **Key takeaways from Nava:**

- Employees value on-demand support and personalized guidance (hence the AI assistant and high satisfaction rates Nava touts).
- HR wants consolidated tools (avoid juggling multiple systems for enrollment, compliance, comms).
- Nava targets 100-1000 employee firms often, which might be a sweet spot for our product too.

To compete, since we may not be a brokerage ourselves (unless we choose to partner with brokers or become licensed), we could offer our platform to other brokers to use (like Ease does) or integrate advisory services in another way. Alternatively, we position as a pure software solution for employers who have a separate broker but want a better platform. Our platform's advanced features (like integrated provider search, extensive integrations, etc.) could appeal to those who want Nava-like tech without changing their broker.

• **Other Notable Competitors:**

- *Zenefits (TriNet Zenefits)*: Zenefits pioneered all-in-one HR + benefits for SMBs, similar to Gusto, though it had compliance issues in the past. Now part of TriNet, it still offers a self-service HR platform with benefits admin. It's competition in SMB tech-focused segment.
- *Rippling*: An emerging player offering a unified platform for HR, IT, and finance. Rippling includes benefits administration integrated with payroll and device management. It's known for a slick UI and modular approach (SMB to mid-market). They emphasize automation between

- systems (like our integration focus).
- *Employee Navigator*: A competitor to Ease, widely used by brokers for SMB and some mid-market. It offers benefits enrollment, HR, and compliance tools. Perhaps less modern UI but very functional. We might encounter it in broker-distribution deals.
- *Benefitfocus*: A long-time enterprise benefits administration platform (recently acquired by Voya). Benefitfocus is used by many large employers and is known for handling complex plans, but it has an older interface. They also connect to carriers and have a marketplace approach.
- *Businessolver*: Another enterprise-oriented benefits administration SaaS (their platform is “Benefitsolver”). They focus on large employers, with strengths in configurable workflows and deep compliance support. They also have a virtual assistant (Sofia).
- *Alight Solutions & Empyrean*: These are service providers with their own platforms, more in the outsourcing realm for enterprises.
- *Nava’s fellow new brokers*: Companies like Mercer’s new platform, Aon’s spin-offs, or other boutique tech brokers (Gather, etc.) might be around.
- *BenAdmin point solutions*: PlanSource, bswift (by CVS/Aetna), and Namely (for mid-sized) are other names.

Each of these has different strengths, but **common features across successful competitors include**: an easy enrollment experience with plan comparisons and cost info, integration with payroll/HR, compliance automation, and value-added services like decision support or concierge. For instance, **Workday’s benefits module** advertises a guided experience with plan comparisons, cost calculators, and provider details in one place – exactly the kind of comprehensive experience we aim to deliver.

- *\*Competitive Gaps and Opportunities:\**

From analysis, some gaps we can target:



- Many traditional enterprise systems (Benefitfocus, bswift) are powerful but **not loved by end-users** due to clunky UI. A modern, mobile-optimized UI with a chat assistant, etc., could beat them on experience.
- SMB-focused platforms like Gusto/Ease excel in simplicity but might lack advanced features (like robust provider search, AI support, or flexible plan configuration for weird scenarios). We can combine *ease-of-use* with *rich features*.
- Broker-led solutions (Ease, Navia) often tie you to that broker's ecosystem. An independent platform that any employer (with any broker) can use might attract those wanting flexibility (some mid-sized employers want to change brokers but keep the same enrollment software, or vice versa).
- **Provider search integration** is not present in most older platforms – they often just link out. This can be a signature feature for us.
- **Real-time integrations** (like API vs batch) – new ones like Rippling highlight integration, but incumbents often still rely on batch files. By emphasizing real-time (or near-real-time) updates (with Noyo or similar), we can claim a technological edge, which brokers/employers increasingly demand (remember, **66% of employers prefer carriers that connect to their software over those with the “best price”** – indicating tech integration is valued).
- **Customer Support and Guidance:** Smaller employers relying on Ease might not get much help beyond their broker; Nava provides lots of support. If we're not a broker, we should still consider a strong customer support model (perhaps partnering with brokers or hiring benefits experts for support) to differentiate on service. A platform + expert hybrid could be attractive.

In conclusion, the competitive landscape is vibrant. Our platform will carve a space by **marrying the user-centric design of modern SMB solutions with the powerful integration and compliance capabilities demanded by larger clients**, and adding unique touches

like integrated provider search and AI-driven support. We should remain aware of evolving features from these competitors (for example, if Ease adds a provider search, etc.) and continuously improve accordingly. The next section discusses how we will differentiate in detail.

## Key Product Differentiators for Market Entry

Entering a crowded market, our platform needs clear **differentiators** – features or capabilities that set us apart from existing solutions. Based on the landscape and user needs analysis, we propose focusing on the following key differentiators:

- **1. Superior Integration and Automation:** Many legacy benefits systems operate with clunky batch processes and siloed data. We will differentiate by offering **real-time or near-real-time data integration** with carriers and HR systems. By leveraging modern APIs (through partnerships like Noyo/Ideon or direct integration), our platform can enable **real-time enrollment updates and automated data synchronization** that older, EDI-bound competitors can't match. This means when an employee enrolls or changes an address, the information flows to carriers and payroll instantly – no waiting for weekly files. The benefit is fewer errors (no latency where data might get out of sync) and a more responsive experience (employees could get ID cards faster, etc.). We will highlight our “integration engine” in marketing, citing that **nearly 60% of employers select carriers based on integration capabilities** – showing we align with that priority. Additionally, automation of tedious tasks (like payroll deduction updates, ACA tracking) will be a selling point: **complex administrative work such as payroll deduction adjustments, ACA compliance tracking, and maintaining consistent records across systems will be automated through our platform's integrations.** This resonates with employers who are stretched thin on HR admin resources.

- **2. Multi-Segment Scalability with Configurability:** Unlike some competitors that only serve either small or large employers, our SaaS is being built to **scale across SMB, mid, and enterprise segments seamlessly**. We will differentiate by allowing a high degree of configuration to meet various needs: for example, a simple 3-step open enrollment for a 20-person company, versus a complex multi-wave enrollment with conditional eligibility and retiree plans for a 5,000-person company. Through a modular design, the platform can “turn on” advanced features as needed (like evidence of insurability forms for life insurance, or retiree billing for enterprise) while keeping the experience clean for simpler cases. Essentially, we can market that our platform grows with you – an SMB can start on it and continue using it as they scale to mid-size, unlike, say, Gusto which they might outgrow. This saves clients from switching platforms as they expand.

- **3. User Experience & Decision Support:** We intend to lead with a **modern, intuitive user interface** that employees and HR actually enjoy using. While this sounds generic, it’s a genuine weakness in many existing platforms. Concretely, our differentiators in UX will be:

- Integrated **provider search and directory** – employees can *immediately check if their doctors are in-network* within the platform, and even filter or get recommendations of plans based on their preferred providers. This is a standout feature that few if any benefits platforms currently offer natively.

- **Personalized plan comparison** – including cost calculators and possibly AI-driven suggestions. We could integrate a recommendation engine in a later phase (like asking a few questions about risk preference or medical usage and suggesting the most cost-effective plan). Competitors like HealthJoy or Nava employ AI for navigation; we can use it at the decision point (e.g., “Which plan might be best for me?” wizard).

- **Mobile-first design** – our platform will be fully usable on mobile devices from the get-go, whereas some older systems are desktop-centric or have limited mobile capabilities. Given many

- employees (especially in industries like retail, manufacturing) may not have a work computer, mobile access is crucial.
- **AI Virtual Assistant (Future Phase):** Introducing a chatbot or virtual benefits assistant within our platform (for example, to answer “What’s my deductible?” or “How do I add a newborn?”) could differentiate in service. We could develop this or integrate an existing one. This aligns with trends (Businessolver’s Sofia, Nava’s AI, etc.) but doing it in an SMB-friendly way could stand out.

In summary, we aim for a **consumer-grade experience** in a business software world. We’ll highlight how our interface “guides employees to make informed decisions with ease” – for instance, Workday brags about that, but many mid-sized employers who can’t afford Workday could achieve it through us.

- **4. Comprehensive Compliance Toolkit:** While ease-of-use is one side, being **strong on compliance** is equally important (especially to win trust of larger clients and brokers). We’ll differentiate by baking in compliance support more thoroughly than SMB competitors do. For example:
  - **Built-in ACA measurement and reporting** – our platform will track variable hours and generate 1095-C forms, where many small biz solutions require an add-on or don’t do it at all. Ease does ACA reporting, but we can make it even more automated and user-friendly (Selerix’s compliance blog gives clues on pain points to solve, like automating look-back calculations).
  - **COBRA workflows** – even if we don’t administer COBRA fully, by flagging qualifying events and preparing notification packets, we show we handle the ugly details. Perhaps partner with a COBRA vendor for seamless service.
  - **Audit logs and data exports** – enterprise-grade features we include by default (so HR can always pull an audit of who changed what, etc.).
  - Emphasize **data security (HIPAA)** – we might pursue HIPAA certification or HITRUST early, showing a differentiator in security

- posture for a young company. If we can say “HIPAA compliant, SOC 2 certified infrastructure from day one,” that alleviates a major concern for clients.
- **5. Speed of Implementation and Carrier Network Breadth:** One pain point for employers/brokers is how long it can take to implement a new benefits system and set up all the carrier feeds. We aim to differentiate by drastically reducing implementation time using modern tech:
  - With API connectivity, adding a new carrier can be faster (Noyo claims new carrier connections activated in days). We’ll strive to have a large library of pre-built carrier integrations and use APIs to avoid waiting on EDI testing cycles when possible. If we can say “get your company up and running in a week” whereas incumbents take 6-8 weeks, that’s compelling.
  - We’ll support **self-service setup** for smaller clients: maybe a wizard to input your plans, or an ability to import from a spreadsheet or from Vericred data. This could let a broker or HR admin configure the platform without heavy IT involvement, making onboarding new customers faster and cheaper. Faster onboarding is a differentiator especially for brokers who might bring dozens of small clients onboard during Q4 renewal season.
  - We can advertise our platform’s wide **carrier network compatibility**. E.g., “integrated with 200+ insurance carriers nationwide.” Ease and others also tout this, but as a new product we need to quickly build that reputation. Using standards and partners helps – we’ll try to support any carrier that accepts 834 (which is basically all) from day one, but we’ll highlight ones we have API connections with as a plus.
- **6. Analytical Insights and Reporting:** Over time, we can set ourselves apart by not just being a transaction system but a insights provider. For instance, providing employers with **analytics on benefits enrollment trends, participation rates, cost projections** etc. Many systems have basic reports, but we could offer a nicer analytics

- dashboard (e.g., which plans are most selected, demographic breakdown of who chose what, etc.). This helps HR demonstrate the value of the platform beyond just compliance – they can make strategic decisions (like if one plan is underutilized, maybe drop it next year, etc.). Some enterprise solutions do this, but bringing such analytics to mid-size clients is differentiating.

- **7. Flexible Service Model:** As noted, one differentiator could be our go-to-market flexibility:

- We might offer the platform **direct-to-employers** who want to buy software and keep their existing broker (many mid-sized companies could fit this).

- We could also offer a **white-label or partner version for brokers/consultants**, unlike say Nava which only uses its tech in-house. If we let regional brokers license our platform to serve their clients, we tap into a distribution channel that Ease currently dominates. Being broker-friendly (e.g., allowing multi-employer broker accounts, agency branding) could win a share of that market.

- Or we combine both: have some direct clients and some via broker partnerships. The idea is we're not stuck in one approach; that agility could itself be a selling point (we aren't in conflict with brokers if we choose not to be; whereas some might view Zenefits or Gusto as competing with brokers because they become broker-of-record).

- **8. Pricing and Value:** While not exactly a feature, pricing model can differentiate. We could adopt a **performance-based or per-employee per-month (PEPM) pricing** that is competitive. For example, some enterprise platforms are expensive; we could offer a tiered pricing that is affordable for SMBs (maybe even free for very small if broker-sponsored, replicating Ease's model) and scaled for larger ones. Additionally, highlighting ROI (like how our automation saves HR X hours, or our decision support helps employees pick cost-effective plans saving claims cost) can differentiate in sales conversations, though it needs data to back it up.

In summary, our platform's differentiators will center around **technology (integration, automation), experience (user-friendly, decision support), and completeness (compliance, analytics)**.

To illustrate, consider an employer comparing our solution to a legacy one:

- With our platform, **carrier connections are faster and errors are rarer** (we can claim e.g. "50% reduction in enrollment errors through automated data checks" based on Noyo's insights). Legacy system might have frequent file errors to clean up.
- Employees on our platform can **find doctors and get help easily**, leading to higher satisfaction, whereas on another system they might have to call HR for help or use external tools.
- Our HR admin interface might consolidate everything (maybe even handle ancillary benefits in one place) which others might not.
- We will also emphasize that adopting our system helps future-proof their benefits administration – because we built it with modern standards, it will adapt to industry changes (like new compliance rules or data exchange formats) more readily than old systems stuck on legacy tech.

Finally, we'll gather client feedback and success stories to refine these differentiators. Early on, focusing on a few key messages – **"More Integrated, More Intuitive, More Intelligent"** – could summarize our edge. For example: *More Integrated* (APIs, real-time), *More Intuitive* (great UX, provider search), *More Intelligent* (AI assistance, smart automation). These are just positioning ideas to guide development priorities.

## MVP Scope and Phased Development Plan

Building a full-featured benefits SaaS is a large endeavor. We will pursue a **phased development strategy**, starting with a Minimum Viable Product (MVP) that covers the core use cases, then iteratively expanding functionality in phases. This approach gets a usable product to market quickly to validate demand and gather feedback, while providing a

roadmap for scaling up to an enterprise-grade solution.

- *\*Phase 0: Research & Design (Pre-development)\*\* – Already in progress.* In this phase, we gather requirements (as in this document), conduct market research, and design the system architecture and data models. We also establish partnerships or select vendors for integrations (e.g., decide on Vericred API, identify top carrier integrations needed). UX design prototypes for the employee and admin interfaces are created and tested with sample users or advisors for early feedback.

## **Phase 1: MVP Development (Core Features)**

- *\*Goals of MVP:\*\** Deliver a working platform that enables a complete open enrollment cycle for a typical employer, with essential compliance and integration features. The MVP should be sufficient to onboard a small or mid-sized client (likely in a controlled beta) and perform enrollment from start to finish, albeit with some manual administrative processes if needed. We will prioritize features that provide the most value and are absolutely necessary for functionality, while deferring more advanced or nice-to-have features to later phases.
- *\*MVP Scope – Features:\*\**
  - **Employee Enrollment Portal (Web):**
    - Secure login for employees (user accounts tied to their email or an employee ID). Possibly implement single sign-on if needed, but for MVP, simple email/password or employer code + personal info to register.
    - Personal info review page and dependent management.
    - Plan selection pages for each benefit category (let's assume MVP focuses on medical, dental, vision – the core health benefits). Ability to compare plans side-by-side.
    - Display of per-paycheck costs and plan details.



- Provider search tool integrated in plan selection flow (this is a standout feature we include in MVP because it significantly enhances decision-making). For MVP, integrate with Vericred's API for a basic search by provider name & zip, and show in-network status for the offered plans.
- Enrollment confirmation and summary view. Ability for employee to download a confirmation PDF or receive an email confirmation.
- Mobile-responsive design so that this can be done on a phone. (A native mobile app could be later phase, but responsive web is MVP).
- **HR/Admin Portal:**
  - HR admin login with access to their company's data.
  - Ability to define company settings (e.g., plan year dates, open enrollment window timeframe).
  - **Plan Setup:** Input the benefit plans offered. For MVP, this can be a form where the admin (or our implementation team) enters plan name, type, coverage tiers, employee cost for each tier, basic benefits info (deductible, etc., maybe as free text or basic fields). We might not build a full plan library database in MVP; it could be manual entry or import. However, to reduce burden, we might at least populate some sample plans or use a template. The admin should also specify which employees are eligible for which plans if there are classes (MVP could assume all active full-timers get same options for simplicity; complex eligibility rules can be Phase 2).
  - **Employee Management:** Admin can add employees or edit their info. (If no HRIS integration yet, this is how employees get in the system – either by import CSV or manual add. MVP should have a CSV import for employee census to save time).
  - **Enrollment Monitoring:** Admin dashboard to see who has enrolled, who is pending, send reminder emails to those who haven't enrolled (even a manual trigger like “send reminder to all not completed”). Also, ability to override or enter an enrollment on behalf of an employee if

- needed (HR often needs to correct something or handle a late entry).
- Basic **reports**: at least an enrollment report summarizing each employee's selections (for payroll deduction and for confirmation to carriers). Possibly an ACA report stub (even if not fully featured, capture that data).
- **Compliance basics**: the system should ensure data needed for 1095-Cs is captured (offer of coverage, etc.). Perhaps generate a simple 1095-C for testing (or at least have data export that could be used to fill one).
- The admin portal should also allow configuration of *dependent verification rules* or similar if needed (MVP might not enforce document upload for dependents – that can be later).
- **Carrier Data Exchange (MVP Simplification)**:
  - For MVP, we will implement **EDI 834 file generation** in at least a basic form. Perhaps focusing on one or two carriers first for testing, then generalize. MVP can output an 834 file (or a set of them) that an admin can download and send to carriers. Alternatively, we set up an SFTP to automatically send to one test carrier. Full automation for all carriers might be phase 2, but showing we can produce a compliant 834 with adds/terms is key.
  - If the above is too heavy for MVP, another approach is to produce simpler reports (like spreadsheet with enrollments) for manual entry to carrier portals as a fallback in initial beta. However, since integration is a selling point, demonstrating at least one automated feed is preferable. We might concentrate on a particular market (say, small group plans in one state) for the beta and build those feeds.
- **Payroll Integration (MVP Simplification)**:
  - Initially, provide a **CSV export of payroll deductions** that HR can upload to their payroll system. E.g., a report listing each employee and their total deduction for benefits per pay period.

- Real-time API integration with payroll (like using something like Finch or a direct ADP API) is likely Phase 2. MVP's manual export still achieves the goal of avoiding manual calculation, albeit with one extra step.

- **Compliance in MVP:**

- HIPAA compliance: ensure our security measures are in place (encrypted DB, https, user access controls).
- ACA: track who is full-time vs part-time (maybe a simple field on employee), track offers of coverage (if an employee waives, mark that). MVP might generate a basic ACA compliance report (like list of FTs and whether they were offered affordable MEC).
- COBRA: MVP can flag if an employee is terminated (status = terminated) and list their coverage that would be eligible for COBRA. We likely won't implement full COBRA workflows in MVP, but have data accessible so HR knows who should get COBRA info.
- Essentially, make sure we're capturing data needed even if MVP doesn't have all automation, so that Phase 2 features can build on it without re-entering data.
- **Auditing & Logging:** Some basic logging of changes (maybe not a full UI for it, but at least record in the database changes like who enrolled in what when – helpful for support if issues arise).
- **Support & Help Content:** Since benefits can be confusing, include an FAQ page or tooltips for common questions (e.g., "What is a deductible?"). Also a way for users to contact support (maybe just a mailto or ticket form for MVP). If we have a small beta group, direct support is manageable; as we scale, we'll integrate a ticketing system or chat.
- **\*MVP Non-Scope (Deferred):\*** To keep MVP lean, we will *exclude* or minimize the following initially:
  - Ancillary benefits like FSA, HSA management, life insurance, disability – MVP might handle these in a very basic way or not at all. Perhaps MVP covers only core health plans. These can be added in Phase 2.

- Advanced eligibility rules (like union vs non-union classes, retiree vs active, etc.) – MVP can assume one set of plans for all or only handle simple division if needed.
- Complex life event processing UI – MVP might not have a self-service life event module. HR can manually make changes for a life event as needed in MVP (like change coverage mid-year). Phase 2 can add an employee self-service QLE flow.
- Fully automated carrier feeds for every carrier – we will build a few in MVP and handle others manually until Phase 2/3.
- API integrations for HRIS – MVP likely will not have direct API integration to HRIS; HR can import/export instead. Phase 2 will introduce e.g. a sync with a system like BambooHR or Workday.
- Mobile app – stick to responsive web for MVP.
- AI chatbot – not in MVP. Possibly Phase 3 after we have sufficient data to train or integrate one.
- Sophisticated analytics dashboard – MVP reports will be functional but minimal; Phase 3 can bring pretty dashboards and predictive insights.
- \*Timeline for MVP:\*\* We estimate ~6 months for a team to build the MVP described (depending on team size and expertise), given the breadth (frontend, backend, EDI integration, API integration). We will likely do iterative internal milestones within that (e.g., basic enrollment flow by Month 3, then integration pieces Month 4-5, etc.). We'll test MVP with perhaps one pilot employer (maybe our own company or a friendly client) in a real open enrollment.

## **Phase 2: Enhanced Functionality and Integrations**

Once MVP is validated (e.g., first client enrolled successfully, feedback collected), Phase 2 will focus on expanding capabilities, automating more processes, and handling more use cases. Likely features in Phase 2:

- **Broader Benefits Support:** Add modules for other benefits like life insurance, disability, FSA, HSA, commuter benefits, etc. This includes building those plan types, contributions, evidence of insurability workflows (e.g., if someone elects coverage over guaranteed issue, alert HR to get approval), integration with providers (maybe link with FSA vendor accounts). This makes the platform more comprehensive rather than just medical/dental/vision.
- **Life Events & Ongoing Changes:** Develop a **Qualified Life Event (QLE) self-service process** for employees. For example, an employee can log in off-cycle and report a life event (marriage, birth, etc.), the system then opens up the relevant benefit changes allowed, and guides them through making changes (similar to open enrollment but just certain plans). Automate effective dating of changes and require documentation upload if needed. This is essential for year-round usefulness of the platform beyond the main enrollment period.
- **Full HRIS/Payroll Integrations:** Implement direct integrations to popular systems. For example, use an API to sync new hires from BambooHR daily, push deduction changes to ADP via API. We might prioritize the systems where our initial clients demand it. Could also integrate single sign-on with common identity providers (Azure AD, Google Workspace) for a smoother login if targeting mid-large employers.
- **Carrier Feed Automation & Library:** Build out a **library of EDI 834 mappings** for a wide range of carriers, and set up routine schedules within the platform (e.g., ability for admin to configure “send enrollment changes to Carrier X every night at 1 AM”). Phase 2 would be when we aim to support most carriers our clients need without manual intervention. We may also integrate with a clearinghouse or use a vendor like Selerix’s integration API or Noyo’s API to handle the back-end of transmission. If Phase 1 proved one or two carriers, Phase 2 generalizes that solution to many.
- **Compliance Automation:**

- Fully implement ACA tracking: ability to define measurement periods, automatically update employee status (ongoing vs new hire measurements), alert when someone should be offered coverage, generate **1095-C/1094-C forms** at year-end ready to file.
- Implement COBRA support or integration: perhaps integrate with a COBRA administration service or at least generate COBRA election notices from the system. Possibly allow COBRA participants to log in and pay for benefits (this might be Phase 3 if complex).
- ERISA document distribution: add a feature to email or post required notices (SBCs, SARs, etc.) to employees via the portal, and track that it was provided (for compliance evidence).
- State compliance: if any clients in states with special rules (e.g., Massachusetts' health coverage reporting), incorporate that.
- **Enhanced Reporting & Analytics:** Add a section for HR to see visualizations of enrollment data, cost projections (like total premium cost to employer vs employee), and the ability to run custom reports (e.g., census reports, enrollment changes over time, dependent lists, etc.). Possibly implement an analytics dashboard using a BI tool integration.
- **User Experience Improvements:** Based on MVP feedback, refine UI. Maybe add a **mobile app** in Phase 2 if user demand is there (or ensure push notifications for enrollment deadlines on mobile). Add more help resources like tooltips for all insurance terms, maybe short explainer videos in the portal.
- **Performance and Scaling:** With more clients, Phase 2 will involve scaling the infrastructure. We'll implement more robust monitoring, perhaps multi-region availability if we have nationwide clients (for reliability). Ensure the system can handle e.g. 10,000 concurrent users if an enterprise client's open enrollment is going on.
- **Security enhancements:** Achieve SOC 2 or similar certification by this phase, implement advanced features like SSO, multi-factor authentication for admin users, IP whitelisting options for clients, etc., to

- satisfy enterprise security reviews.

Phase 2 essentially takes us from a functional product to a more **feature-complete competitive product**, suitable for a wide range of clients and with efficiency gains (less manual work for our ops and for clients).

## Phase 3: Advanced and Differentiating Features

Phase 3 and beyond will focus on truly differentiating long-term features and further scalability:

- **AI and Decision Support:** Introduce an AI benefits advisor (chatbot) within the platform. Leverage our growing dataset to train it to answer common questions (“What’s the difference between my plans?”, “How much have I paid toward my deductible?”, etc.). Possibly integrate with healthcare cost databases to give personalized estimates (like Castlight-type functionality for cost of procedures, or help choosing a plan based on predicted total cost given an employee’s conditions).
- **Healthcare Navigation Integration:** Partner or build features for year-round healthcare support, moving somewhat into HealthJoy’s territory. For example, after enrollment, the platform could offer a “Find Care” service: search for providers (which we already have), or connect to telemedicine, or show how to utilize benefits (like push reminders for free preventive checkups).
- **Market Expansion:** Add support for more types of benefits or arrangements:
  - Include support for **self-funded plan administration** (if we target bigger companies, maybe integrate with TPAs or ingest claims data to show employees their claims or accumulate deductibles—though that’s more on carrier side).
  - Integrate **voluntary benefits** offerings (like accident insurance, pet insurance, etc., possibly through partnerships).

- Consider a **marketplace model** where employers can browse and select new benefit offerings (some platforms like Gusto or Zenefits have marketplaces for things like financial wellness, we could curate add-ons).
- **Global or Additional Modules:** If we ever expand beyond US, add country-specific benefits modules (but that's a big expansion likely beyond near-term). Or expand into adjacent HR functions like leave management just to add value (some benefits systems do touches of that).
- **Broker Portal:** If not earlier, by Phase 3 we definitely want a multi-client broker/agency portal if we partner with brokers. This allows a broker to log in and oversee all their client employers in one place, helping with their enrollments, etc. It's crucial to penetrating the broker channel.
- **White-label and Customization:** Provide options for clients to theme the portal with their branding, or integrate our platform's functionality into their existing intranet via APIs/embeddable widgets. This could differentiate for large clients who want a seamless brand experience.
- **Improved Carrier Connectivity:** Possibly by Phase 3 we rely less on EDI entirely and move to direct APIs or real-time connectivity for most transactions, thereby achieving the vision of instant updates. This could include using emerging standards like the **FHIR-based enrollment standards** if they develop, or deeper partnerships with carriers to be in their ecosystems.
- **Data-driven Insights for Cost Control:** Provide analytics to employers like "if you shifted X% of employees to Plan B you'd save Y" or identify utilization patterns (requires claims data access typically, which might be out of scope unless partnering with carriers or wellness programs). We could position the platform as not just administrative but also helping contain costs (which appeals to CFOs and benefits managers trying to strategize).



- **API as a Product:** Open up our own APIs so that other vendors or client IT departments can pull data or integrate the platform into a bigger IT ecosystem (for instance, allow an enterprise's data warehouse to query our system for enrollment data, or allow a mobile app to use our endpoints). This increases stickiness and differentiation as a platform versus a closed system.

Each of these advanced features would be driven by customer feedback and market demand as we grow. Importantly, we will continuously iterate with input from actual users (employees and HR) to polish features, ensuring we maintain an intuitive experience even as complexity grows.

- **\*Phased Rollout & Testing:\*\*** For each phase, we will plan beta testing. MVP (Phase 1) with a friendly client or internal test, Phase 2 features with a subset of clients who volunteer to try new features early, etc. This ensures quality and that we build the right things.
- **\*Resource Planning:\*\*** MVP can be achieved with a smaller team; Phase 2 and 3 will require scaling the development, QA, support teams accordingly. We'll likely invest in a client implementation team by Phase 2 to help onboard new customers (especially for larger ones with more complexity).
- **\*Market Entry Strategy (in tandem with dev phases):\*\*** We might initially onboard a handful of clients manually (high-touch) to ensure success. As features stabilize by Phase 2, we could accelerate sales (direct and via brokers). By Phase 3, we should have a mature platform that can be marketed widely as a top-tier solution.
- **\*Actionable Recommendations Summary:\*\***
- **Start with core high-impact features** (enrollment workflow, basic integrations) to prove value quickly.
- **Leverage third-party APIs** (Vericred, etc.) in MVP to accelerate development of complex features like provider search.
- **Incrementally automate:** don't wait to have every integration perfect; deliver some in MVP, automate more in Phase 2, so users see

- continuous improvement.
- **Gather user feedback** at each phase and iterate – e.g., survey employees after using the MVP for open enrollment, use that to prioritize Phase 2 enhancements.
- **Keep compliance up-to-date:** even in MVP, build with changing laws in mind (e.g., code ACA logic flexibly), so we don't paint ourselves into a corner.
- **Invest in UX and support** as differentiators: continue to refine UI and plan to add AI/chat support by Phase 3 when we have enough data.
- **Scalability:** Ensure from the start that architecture (multi-tenancy, cloud infra) can handle onboarding many clients in Phase 2+ without re-engineering.
- **Marketing:** Highlight differentiators (integration, experience, compliance) as they come online. For instance, once Phase 2 gives full integration, actively promote that advantage to win clients from competitors who use batch processes.

By following this phased plan, we mitigate risk and funding by delivering value early, and we lay the foundation for a robust platform that can truly compete with the best in the market as we progress through each stage.

- --
- Sources Cited:\* The information and strategies above are informed by industry data and best practices, including insights from benefits technology providers and studies. For example, the importance of broker integration and carrier connectivity is underscored by a LIMRA study where **59% of employers select carriers for their ability to connect to benefits tech**, and modern ben-admin platforms are **using APIs to enable real-time enrollment, cutting errors and costs versus legacy EDI processes**. The plan also reflects features proven valuable by competitors, such as **user-friendly plan comparison and decision support tools to improve the employee experience**, and integrated provider data which **guides employees**

- **toward plans including their preferred doctors.** These references, along with others listed, provide a factual backbone for the recommendations in this plan.