
Castlight

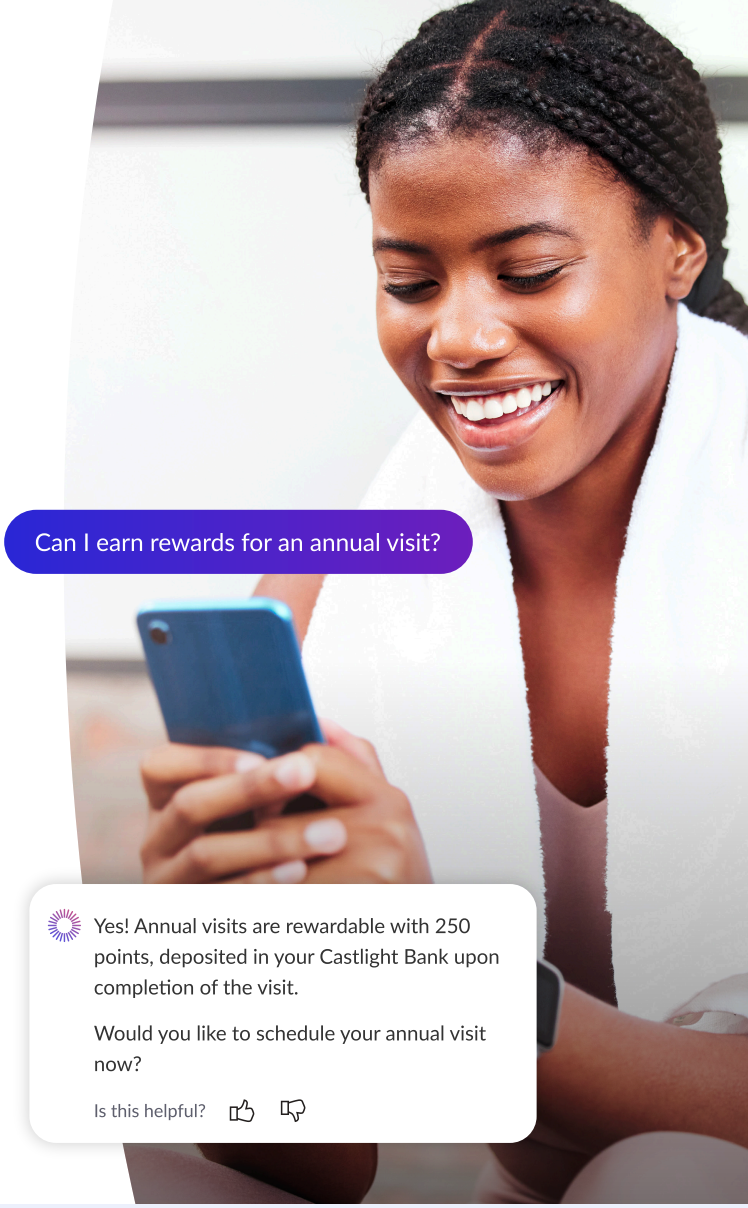
# Designing & Securing AI in the Castlight Platform

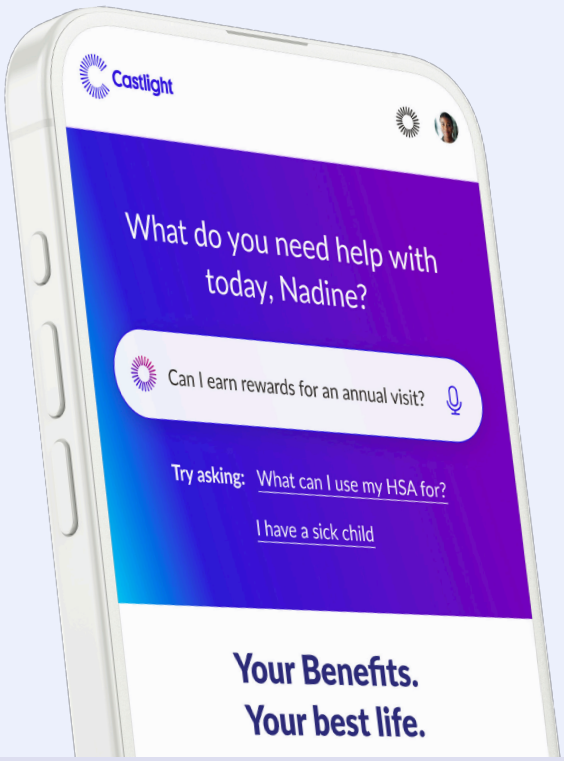
## Introducing Prism

At Castlight, we're committed to building AI products safely, responsibly, and ethically. Our platform is designed with security, privacy, and compliance at the forefront. We understand that security and privacy are paramount concerns when adopting new technologies, especially those involving sensitive data.

Our newest feature within the Castlight platform, Prism, invigorates healthcare navigation by providing users with seamless access to their healthcare information and an all-inclusive search experience. Prism intelligently navigates on users' behalf by leveraging conversational AI. Users can ask anything related to benefits, healthcare, or well-being and Prism will deliver quick and accurate answers. In addition to answering many questions directly, Prism provides links to relevant benefits or care-related content and can transfer conversations seamlessly to Care Guides for more personalized assistance all within a single, integrated chat experience.

In this guide we break down what you need to know about Prism, including how data is accessed and used, frequently asked questions, and more.




## Prism Features

Prism helps users navigate their personalized healthcare journeys through the use of AI. It's capabilities are continually expanding and has the ability to do many tasks that can be performed throughout the application all within a natural language chat experience. These features include finding providers, scheduling appointments, viewing care teams, answering questions about health plan details and benefit programs available, and much more.

Importantly, we take customer privacy and security very seriously and have posted an extensive Artificial Intelligence Transparency Statement on our Trust Portal that details the capabilities of Prism, how we keep customer data safe, and how we strive to ensure that Prism is a fair tool for all.

## How Does Prism Work?

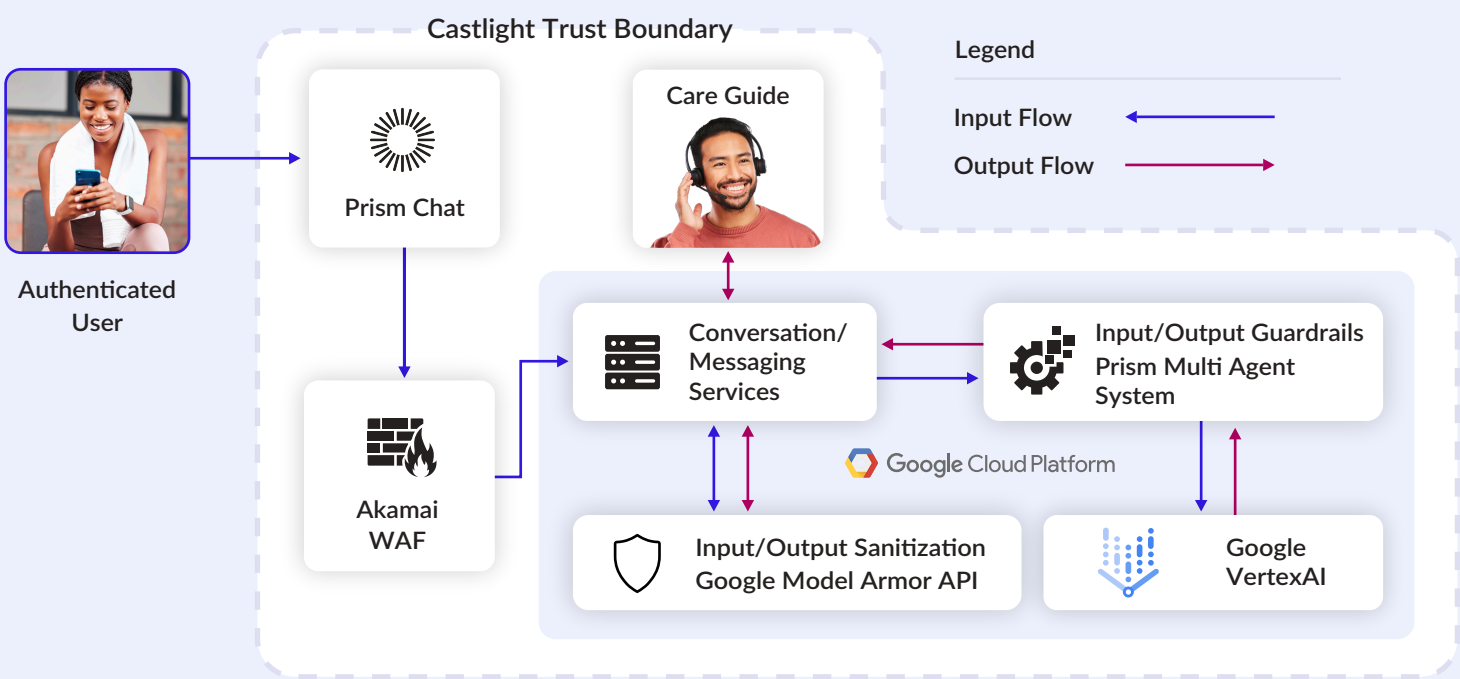
Prism is a multi-agent AI system that uses large language models in conjunction with many data sources to help users navigate their healthcare needs. Information input by users into Prism is never used to train models or retained anywhere outside of the Castlight trust boundary. Keeping customer data secure, ensuring user privacy, and not providing medical advice were of utmost importance when designing Prism. For this reason we have multiple redundancies for safety checks throughout the system.

Specifically, when a user asks a question, we carry out a series of input guardrail checks to ensure that the request isn't malicious. Once passing that check, the system information includes clear instructions not to share medical advice. Lastly, once the system has constructed an answer we have a final node that provides a final output guardrails check to ensure that Prism has answered the user's question satisfactorily and isn't sharing any information or advice that it shouldn't be sharing.

Prism undergoes extensive testing with malicious requests, actual relevant commonly asked questions, and many ways of requesting medical advice to ensure that it responds appropriately. These tests are conducted regularly during development and also periodically in production.

The tests executed periodically in production are part of a self-hosted observability, monitoring, and alerting platform that allows us to check for a broad range of issues in real-time and respond accordingly. For example, if Prism isn't providing relevant answers our LLM-as-a-Judge tests will pick up on this and we can efficiently investigate potential causes. Similarly, the system provides monitoring and alerting for hallucinations, medical advice, toxicity, and many other evaluations.

Prism has the ability to lookup health plan details, benefits, claims, and other PHI and PII that are specific to the user when answering questions. To accomplish this, it uses our existing APIs to pull these data into the session using the same advanced encryption and authentication mechanisms used throughout our application.



# Quality of Prism

We employ a comprehensive, multi-layered approach to guarantee the quality, safety, and reliability of Prism. Our strategy integrates established software development best practices with a robust framework for continuous oversight, real-time monitoring, and human-in-the-loop (HITL) review.

## Integration with the Software Development Lifecycle (SDLC)

Prism is developed and maintained within our established SDLC, ensuring that foundational quality standards are met at every stage. This includes:

- Code Review: all changes undergo peer review to verify logic, security and adherence to coding standards.
- Manual and Automated Testing: comprehensive testing suites are utilized including unit, integration and end-to-end test alongside dedicated manual quality assurance.
- Smoke Testing: core functionalities are verified via automated smoke tests to confirm system stability and readiness.

## Human-in-the-Loop (HITL) Oversight

### Realtime Monitoring

We've implemented a real-time Human-in-the-Loop monitoring that provides visibility into Prism's performance. This includes:

- Observability and tracing functionality that provide granular visibility into each interaction. This allows our team to benchmark and align model performance.
- Foundational functionality monitoring tracks the accuracy of Prism, tone and sentiment scores, and guardrail compliance.
- Active feedback mechanisms allow HITL operators to monitor response satisfaction scores from users and leverage annotation functionality to flag potential issues, edge cases, or areas for improvement.
- Proactive alerting notifying our engineering teams if critical quality or safety issues are identified.

### Careguide Transfers

For customers leveraging our Care Guide offering, users have the option to be transferred to a Care Guide at any point during a conversation with Prism. When transferred, the Care Guide receives a summary of the conversation with Prism to ensure a seamless, optimal user experience.

### Retrospective Quality and Continuous Improvement

Beyond real-time monitoring, our Customer Service Center of Excellence Quality Program conducts systematic retrospective analysis to drive continuous model and process improvement.

A sample of Prism interactions are manually reviewed by our team to assess performance against defined criteria cover accuracy, tone/sentiment, and compliance with guardrails. The reviews provide insights into error trends, escalation to Care Guide volumes and areas requiring fine-tuning or guardrail reinforcement.

### Quality Gate Failures and Rollback Process

Our final safeguard is a defined procedure for handling significant quality degradations via rollback. If retrospective sampling or real-time monitoring indicates a Quality Gate Failure – a breach of a critical performance or safety threshold – we initiate our defined rollback process. This reverts the production environment to the last known stable and verified version of Prism or turns off the Prism feature altogether, containing the incident and allowing for root-cause analysis.

## Securing Prism

At the core of Prism's capabilities is a multi-layered security framework designed to protect your data and ensure reliable operation. We understand that trust is paramount, especially in healthcare, and our approach combines industry best practices with advanced Google Cloud security features.

### Secure-by-Design Architecture on Google Cloud

Prism is hosted and deployed within Google Cloud Platform (GCP) using Google's VertexAI and GenAI APIs. Our security architecture adheres strictly to Google Security Foundations 2.0. This framework provides a comprehensive set of security best practices, configurations, and controls specifically designed for building secure applications on GCP.

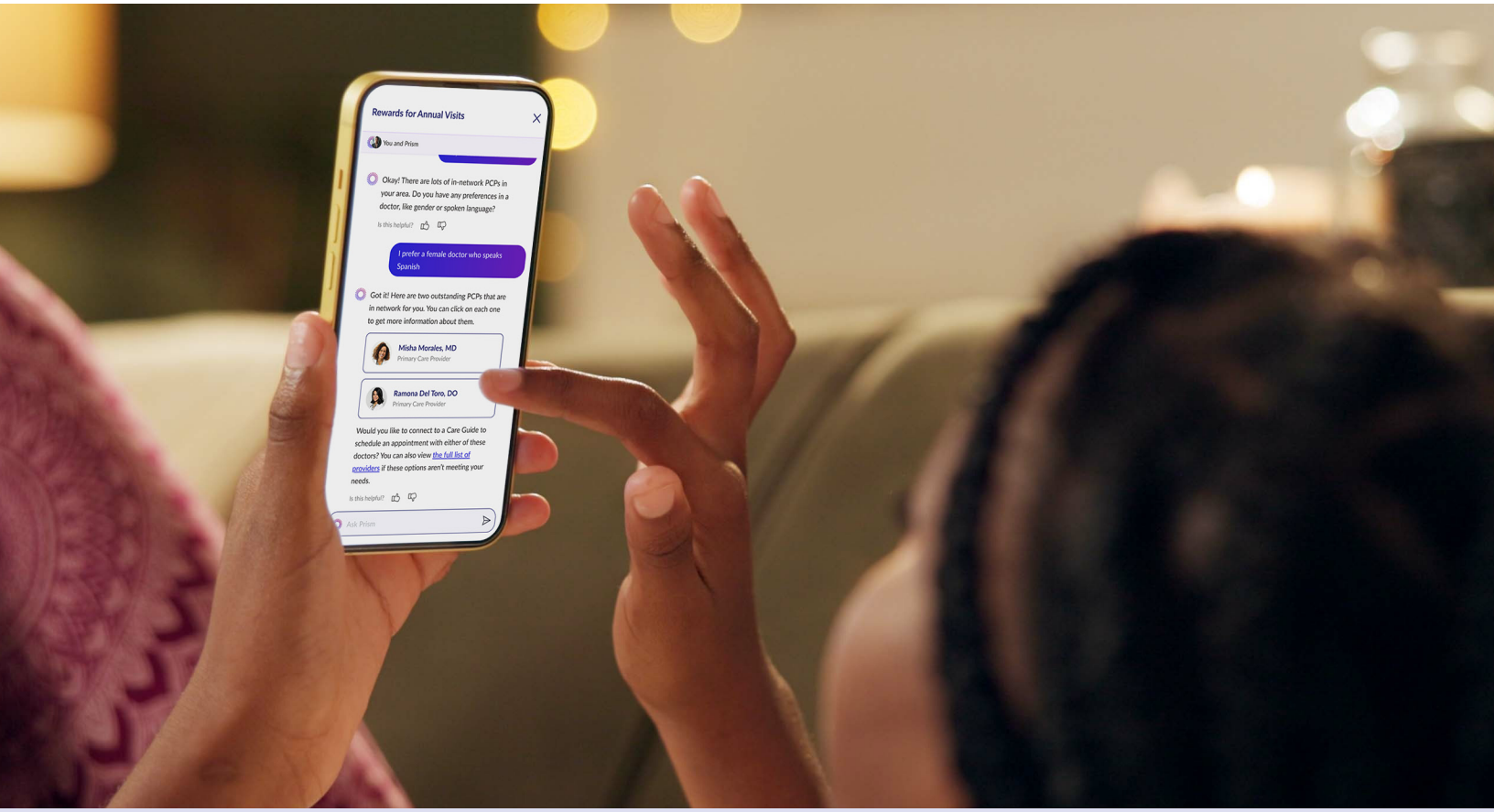
All data within Prism and its integrated GCP services is encrypted at rest using customer managed encryption keys (CMEK) and in transit using TLS 1.2+.

### Runtime Protection with Google's Model Armor

To provide comprehensive security for interactions with VertexAI and GenAI APIs, Google Cloud's Model Armor is deployed as a dedicated security layer in front of Prism. Model Armor is a fully managed Google Cloud service designed to act as an AI firewall, screening both user input and the outputs in real-time for AI specific vulnerabilities such as malicious content, prompt injection, jailbreak, malware and content safety.

### Built on Existing LLM Standards

Prism's development and deployment adhere to recognized security standards and frameworks for LLMs, including those from OWASP LLM Top 10 and MITRE. This foundational approach ensures that common vulnerabilities are identified and addressed proactively.



## Mapping to the OWASP LLM Top 10

| OWASP LLM Risk | Attacker Goal | Our Mitigation Strategy |
|---|---|---|
| **LLM01: Prompt Injection** | Extract Sensitive information, Unauthorized access to data, alter agent or LLM behavior | • Constrain model behavior via system prompt<br>• Define and validate output formats<br>• Filter input/output via quality agent and intent classifier agent<br>• Filter input/output via Model Armor<br>• Enforce least privilege access<br>• Human in the loop<br>• Adversarial attack testing |
| **LLM02: Sensitive Information Disclosure** | Leak PHI & PII Information | • Sanitization via quality agent<br>• Input/Output validation with Model Armor<br>• Strict access controls<br>• Restricted data sources and session isolation<br>• Transparency in data usage |
| **LLM03: Supply Chain Security** | Gain unauthorized access, steal sensitive information, alter agent behavior | • No public datasets in use<br>• Model maintained and vetted by Google<br>• Vulnerability scanning executed in deployment in pipeline<br>• Signed artifacts deployed that cannot be modified during runtime<br>• AIBOM defined in code |
| **LLM04: Data and Model Poisoning** | Corrupt Agent and Model Behaviour, instruct to make mistakes | • Google's VertexAI platform does not train on nor retain data<br>• Robust, immutable logging and monitoring in place<br>• Defined test datasets used to benchmark agent behavior and detect drift<br>• Least privilege access controls in place on the vector embedding database to prevent tampering<br>• Defined incident response and context purge plan |
| **LLM05: Improper Output Handling** | Exploit weaknesses in validation and sanitization of outputs leading to XSS, CSRF, SSRF, privilege escalation, or remote code execution | • Constrain model behavior via system prompt<br>• Define and validate output formats<br>• Filter input/output via quality agent and intent classifier agent<br>• Filter input/output via Model Armor<br>• SAST scans detect improper output encoding<br>• Adversarial attack testing |
| **LLM06: Excessive Agency** | Deviate from intended system behavior | • Tool filtering in place to only allow minimum necessary tools for intended operation<br>• Tool permissions are managed and defined outside of Prism<br>• Strict access controls following least privilege<br>• Session isolation with restricted data sources |
| **LLM07: System Prompt Leakage** | Learn internal prompt logic, bypass filters or guardrails by reverse engineering instructions | • Input/Output sanitization via quality agent<br>• Input/Output sanitization via Model Armor<br>• Session isolation<br>• Adversarial attack testing |

## Mapping to the OWASP LLM Top 10 (cont.)

| OWASP LLM Risk | Attacker Goal | Our Mitigation Strategy |
|---|---|---|
| **LLM08: Vector and Embedding Weaknesses** | Unauthorized data access, context leaks, or manipulated model outputs | • Quality agent assesses response for coherence and policy adherence<br>• Permission aware RAG<br>• Adversarial attack testing |
| **LLM09: Mis-information** | LLMs produce false or misleading information (hallucinations) that appears credible, leading users to make incorrect decisions | • Cross-Verification and Human Oversight:<br>• Human in the loop<br>• Quality agent assesses response for coherence and policy adherence<br>• User feedback feature to rate Prism's responses |
| **LLM10: Unbounded Consumption** | Exhaust token/quota limits leading to excessive latency or denial of service (DOS) | • Input validation (length validation)<br>• Backend rate limiting<br>• Web application firewall rate limiting<br>• Availability and uptime monitoring |

## FAQ

**Is our data used to train the LLM?**
No. Prism uses Google's VertexAI as the foundational platform for the LLM. We leverage Google's enterprise licensing for VertexAI to ensure that Google never trains on or retains any of our users' data.

**How is PII/PHI handled overall between the user input and response generation?**
PHI/PII is protected by the existing Castlight systems. Prism uses the same platform architecture, modeled using the Google Cloud Security Foundations and adheres to the same policy and data storage standards as the entirety of the Castlight platform.

**Does your AI solution ensure full HIPAA compliance, specifically regarding the handling, storage, and transmission of PHI?**
All features in the Castlight platform are HIPAA compliant and HITRUST / SOC2 certified. Our organization has a business associate agreement with Google and Google's entire product suite including VertexAI is also HITRUST / SOC2 certified and HIPAA compliant.

**Has penetration testing been performed on this feature?**
Yes, both internal and third party penetration testing have been performed against Prism.

**Are any third party integrations or applications used in support of this feature?**
All data stays within the Castlight trust boundary. We have business associate agreements with both Google (used for infrastructure and VertexAI platform) and Salesforce (used to connect with a Care Guide from Prism).