vFunction Architectural Observability Manager

Shift Left for Architects

vFunction **Architectural Observability Manager** enables architects to assess, monitor, find, and fix application modernization issues before they result in a technical debt disaster - the first "shift-left" product for architects. The first application modernization solution to continuously monitor, baseline, detect and isolate critical application architecture anomalies.

Catch Architectural Drift Anomalies Before Disaster Strikes

Baseline	Monitor & Detect	Pinpoint
Application Architecture	Architectural Drift	Architectural Anomalies

Architectural Observability Prevents Meltdowns

Architectural meltdowns and technical debt disasters have become all too commonplace. Technology leaders must move to a continuous modernization culture where technical debt is managed by setting organizational baselines for technical debt, monitoring for architectural drift, and catching major architectural anomalies before they become business catastrophes.

vFunction **Architectural Observability Manager** enables software architects, developers, and engineering leaders to both find and fix application or microservice architecture anomalies, determine architectural baselines, set thresholds, and alert when critical modernization anomalies are detected.

Unchecked architectural drift causes:

- Ballooning technical debt
- Architectural erosion
- System failures
- Security breaches
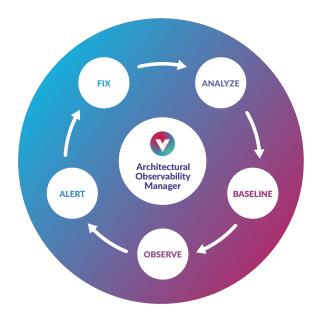- Sky-high cloud compute costs
- Dead code creep

vFunction Architectural Observability Manager

Find and Fix Architectural Anomalies

Observability-driven architecture consists of a repeatable best practice that manages, observes, and fixes application architecture anomalies on an iterative, continuous basis.

- Baseline, monitor, and alert on architectural drift issues such as the detection of new services, new common classes found, service exclusivity changes, new dead code uncovered, new high debt classes identified and more.

- Get notified of architectural events immediately through various alert systems including Slack, email, and the vFunction Notifications Center.

- Once an architectural events are pinpointed, address them directly with your development team or use vFunction **Refactoring Engine** to resolve the issues.

Analyze

Analyze the application architecture in production, test, or staging with dynamic and static analysis.

Baseline

Set architectural baselines for technical debt, detected classes, dead code, common code, and high debt classes unique to your application.

Observe

Observe and actively monitor for architectural drift from the established baselines to detect anomalies indicating significant architectural erosion events.

Alert

Detect and pinpoint architectural events and threshold violations - often down to the offending Java class - and receive alerts in Slack, email, or the vFunction Notification Center to determine appropriate remediation.

Fix

Triage architecture anomalies and re-architect or refactor issues identified, including extracting services with vFunction Refactoring Engine

FIX → ANALYZE

Architectural Observability Manager

ALERT

BASELINE

OBSERVE

vFunction

Request a Demo