RESILIENT OPERATIONS · AIOPS

# Service Topology & Dependency Mapping

A continuously learned model of how every service, API, data store and infrastructure component depends on the next — the context AIOps, powered by iStreet Network, uses to correlate signals and isolate root cause.

## Overview

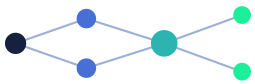
Alerts describe symptoms in isolation. When applications, microservices, shared databases and network paths are monitored separately, each team sees its own layer, and the relationship that carries a failure from one service to the next stays undocumented. Service Topology & Dependency Mapping establishes that relationship as a live model: what connects to what, which services sit downstream, and what else is at risk when one component degrades.

## How AIOps Uses Service Dependency Mapping

### 01 Ingest

- Topology and infrastructure relationships
- Application and microservice data
- Network flow, API latency, database and storage telemetry

### 02 Map



- Machine-learned service dependency graph
- Built from live production traffic
- Refreshed in near real time as services deploy and change

### 03 Reason

- Correlates related alerts by known dependencies
- Traces symptoms to the root cause
- Identifies impacted downstream services

### 04 Act

- Routes the incident to the team that owns the root-cause service
- Prioritises by SLA exposure and business impact
- Triggers healing workflows, with approval gates for high-impact actions

An alert on its own carries no service context. Service dependency mapping tells AIOps which services run on the affected component, which services depend on them and which business transactions flow through that chain. That context is what turns isolated alerts into one correlated, prioritised incident.

## Key Capabilities

### Continuous discovery

Learns dependencies from live production traffic and refreshes the graph as services deploy and call patterns change.

### Layered service map

Services grouped into application, cache, load-balancer and custom layers, with entry points and external callers marked.

### Hybrid and Kubernetes

Kubernetes services discovered automatically and mapped beside monolith and on-premises services in one topology.

### Impact and blast radius

Traces incoming and outgoing dependencies for any service to show what sits downstream of a degradation.

### Topology-aware RCA

Follows symptoms across the graph to the component, service and parameter most likely driving an incident.

### Dependency-based correlation

Groups alerts by known service relationships, not time proximity, and routes incidents to the owning team.

TECHNICAL SPECIFICATIONS

# What the Dependency Model Covers, and How It Runs

| SPECIFICATION | DETAIL |
|---|---|
| **Discovery & Modelling** | |
| Discovery method | Machine-learned from production traffic patterns; graph refreshed in near real time |
| Service model | A service is a logical cluster of load-balanced instances performing one function |
| Kubernetes | Automatic service discovery; sidecar-based request monitoring for each container |
| Hybrid coverage | Kubernetes, monolith and on-premises services in one application topology |
| Telemetry inputs | Topology relationships, application and microservice data, network flow, API latency, database and storage telemetry |
| **Map & Visualisation** | |
| Map scope | One service map per organisation, region or tenant account (multi-tenant service providers) |
| Node types | External entities, entry-level services and service layers; custom layers and icons supported |
| Navigation | Search by full or partial name; zoom to fit; map overview; saved layout alignment |
| Dependency view | Highlights the incoming and outgoing connections of any selected service |
| Service dashboard | Health summary, workload and behaviour KPIs, active signals and events per service |
| **AIOps Integration** | |
| Event correlation | Alert grouping by service dependency across infrastructure, applications, network, logs and ITSM |
| Root-cause analysis | Topology-aware isolation with timeline-based fault reconstruction |
| GenAI investigation | Conversational dependency queries through Talk to Incident |
| Workflow | ITSM, runbook automation and CI/CD integration; human-in-the-loop approvals with audit trails |
| **Platform & Deployment** | |
| Scale | Designed for millions of events per minute across hybrid, multi-cloud and legacy environments |
| Deployment | Sovereign and data-resident by architecture; on-premises and local-first options |

## Operational Outcomes

### Faster Mean Time to Identify (MTTI)

Helps teams quickly determine whether their services are involved in the failure path, reducing unnecessary investigation.

### Reduced Mean Time to Resolution (MTTR)

Accelerates incident investigation by identifying service dependencies, tracing failure paths, and identifying potential root causes.

### Improved SLA Management

Enables incident prioritisation based on affected services, dependency criticality, and business impact to support SLA commitments.

## About iStreet

iStreet Network Limited (BSE: 524622) delivers an enterprise-grade sovereign AI ecosystem for regulated industries. Through its Resilient Operations solutions, iStreet enables dynamic service topology and dependency mapping across complex IT environments.

Using operational telemetry, the solution continuously discovers and maps dependencies across applications, microservices, infrastructure, databases, networks, and business services. This provides operations teams with the context needed to correlate events, understand service relationships, assess downstream impact, and isolate probable root causes.

Service Topology & Dependency Mapping is part of Sanjeevani of AI™, iStreet's AI Centre of Excellence and integrated framework that brings together observability, security, governance, risk, and compliance capabilities within a unified architecture. The Resilience Operations Centre (ROC) operationalises these capabilities, enabling teams to monitor service health, investigate incidents, and coordinate resolution across interconnected IT environments.