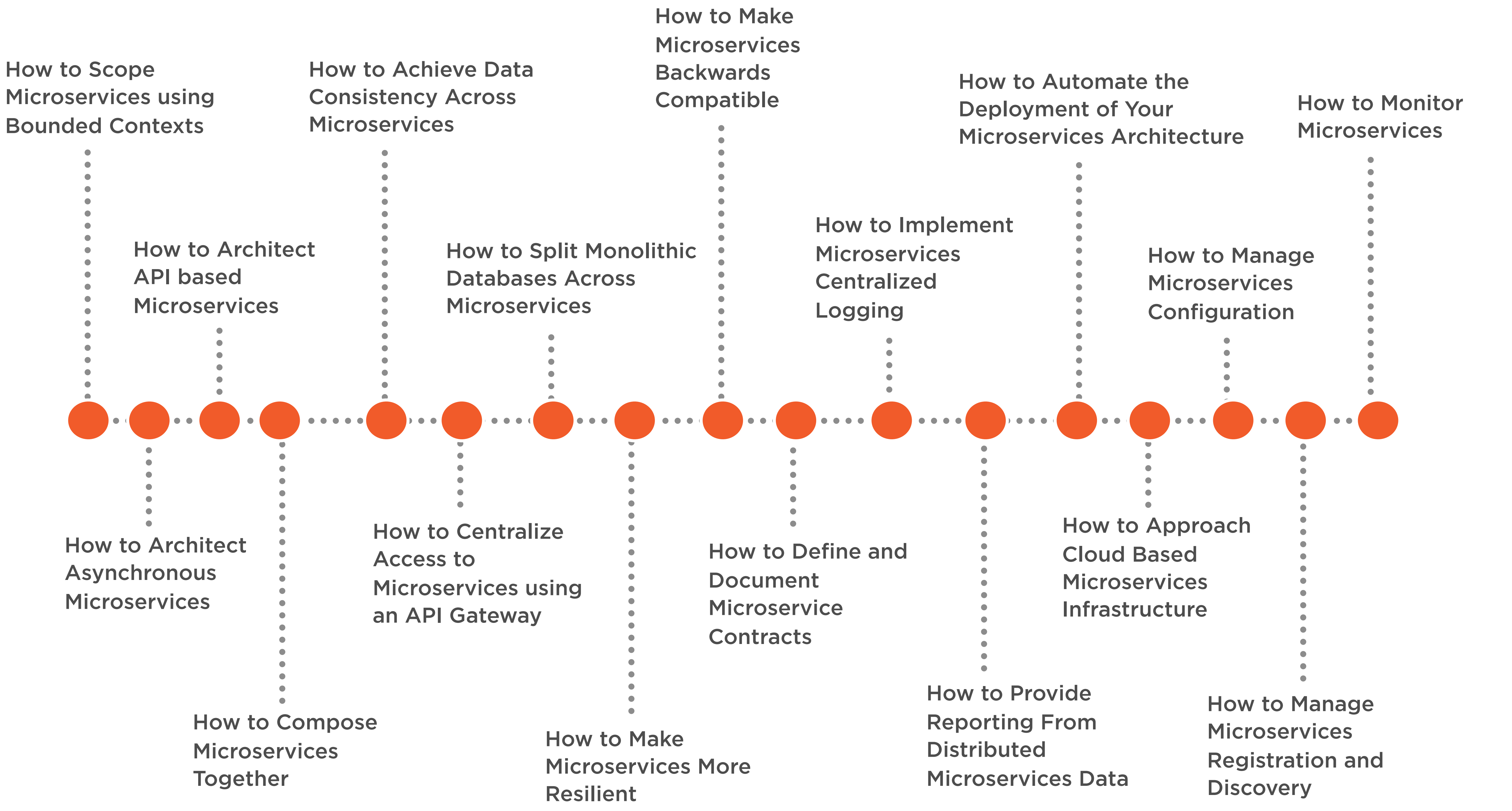
How to Monitor Microservices

Rag Dhiman

@ragdhiman www.ragcode.com

Microservices Architectural Design Patterns Playbook



How to Scope Microservices using Bounded Contexts

How to Architect Asynchronous Microservices

How to Architect API based Microservices

How to Compose Microservices Together

How to Achieve Data Consistency Across Microservices

How to Centralize Access to Microservices using an API Gateway

How to Split Monolithic Databases Across Microservices

How to Make Microservices More Resilient

How to Make Microservices Backwards Compatible

How to Define and Document Microservice Contracts

How to Implement Microservices Centralized Logging

How to Provide Reporting From Distributed Microservices Data

How to Automate the Deployment of Your Microservices Architecture

How to Approach Cloud Based Microservices Infrastructure

How to Manage Microservices Configuration

How to Manage Microservices Registration and Discovery

How to Monitor Microservices

Microservices Architectural Design Patterns Playbook

Microservices Architecture

Rag Dhiman

@ragdhiman www.ragcode.com

Microservices Architectural Design Patterns Playbook

Rag Dhiman

@ragdhiman www.ragcode.com

Overview

Microservices Monitoring

Monitor Key Metrics

Monitor SLA Metrics

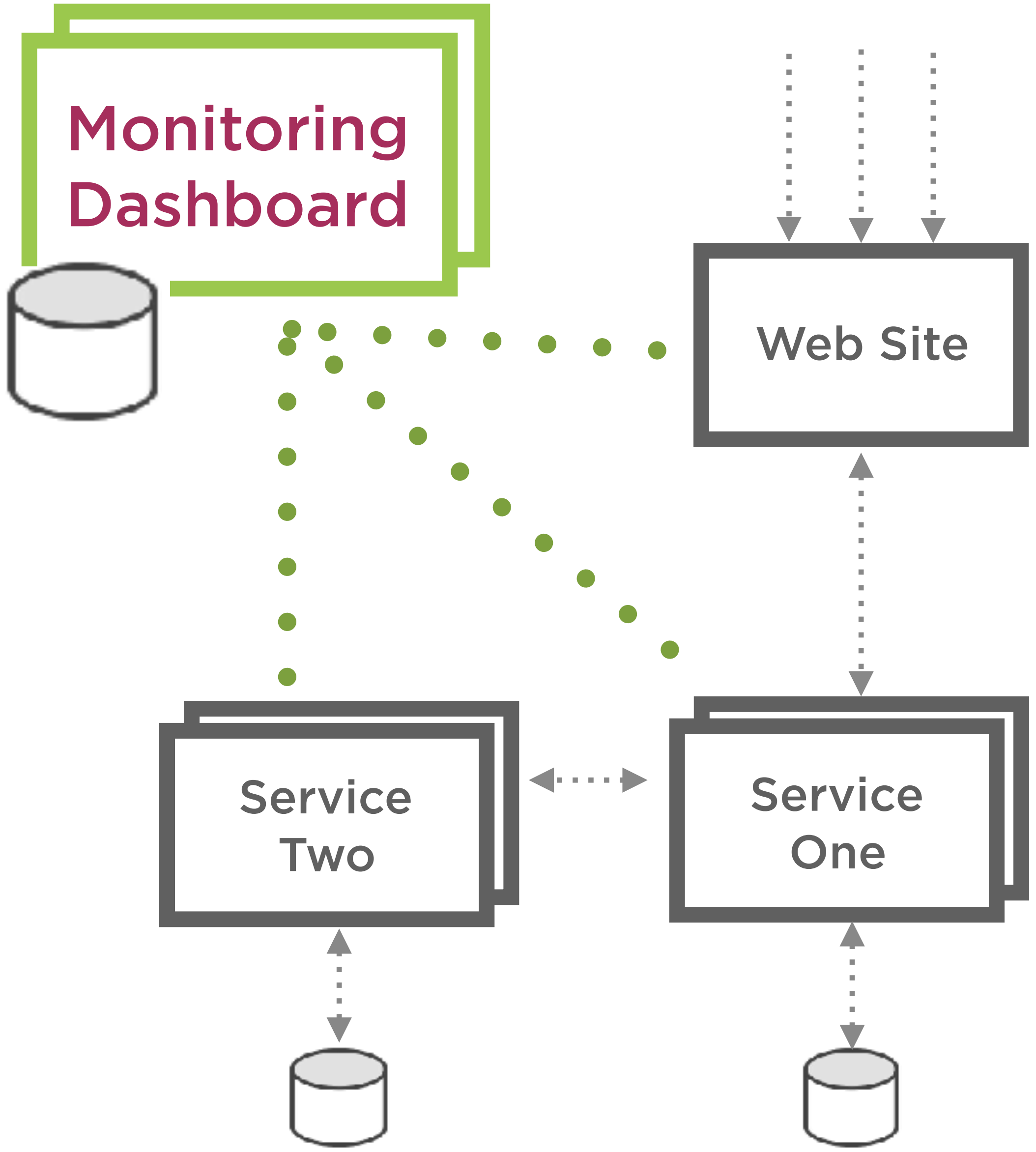
Monitoring Dashboards

Alerting and Monitoring

Defining Thresholds for Alerts

Monitoring Tools and Patterns

Microservices Monitoring



Why the need to monitor

- Microservices complexity
- High availability is a requirement

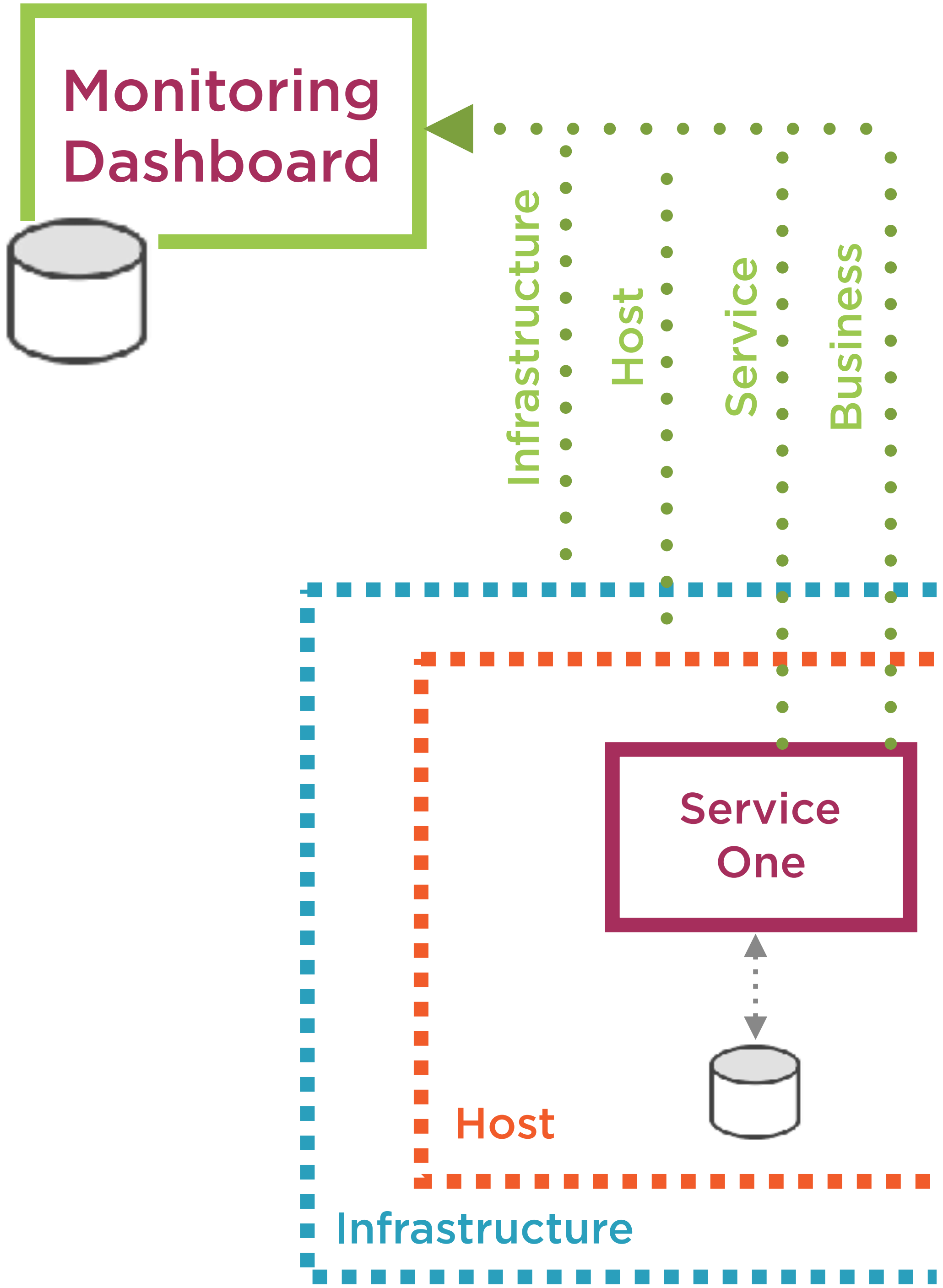
Causes of microservices outages

- Deployment issues
- Delayed and long repair
- Downstream dependency issues

Monitoring is the proactive solution

- Know the state of all components
- Know the state at every stage
- Collect all statistics using metrics

Monitor Key Metrics



Host and infrastructure metrics

- State of hosts and infrastructure
- State affects microservices

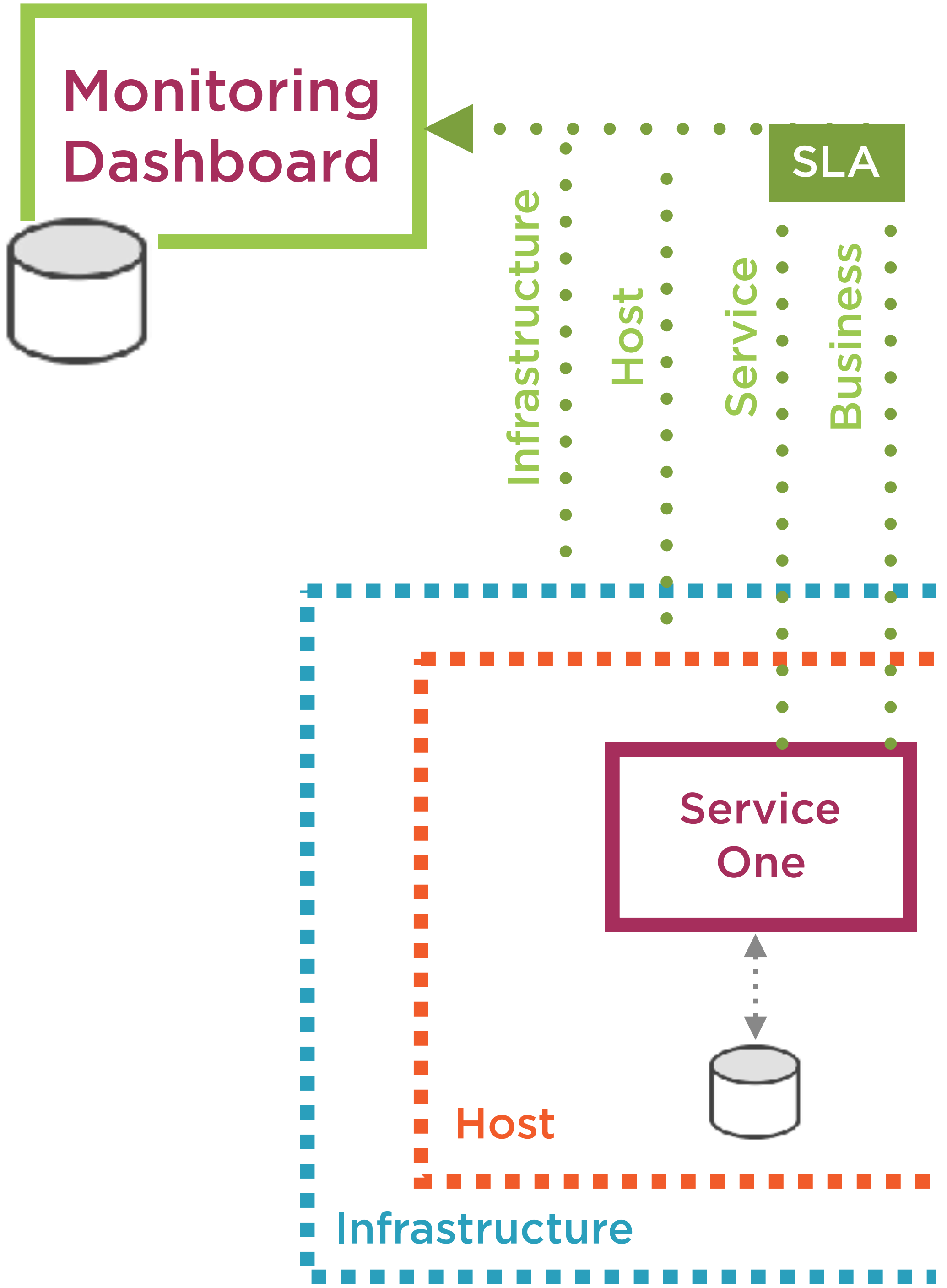
Monitor microservice metrics

- Availability
- Response rate
- Success or failures of endpoints
- Errors, exceptions and timeouts
- Health of dependencies
- Metrics at different stages

Business metrics

- Hidden issues and failures
- Sudden changes in behavior

Monitor SLA Metrics



Monitor SLAs at all levels

- Service level
- Service endpoint level

Examples of SLA metrics

- Throughput and uptime
- Concurrent clients
- Average response times

Impact of dependencies on SLAs

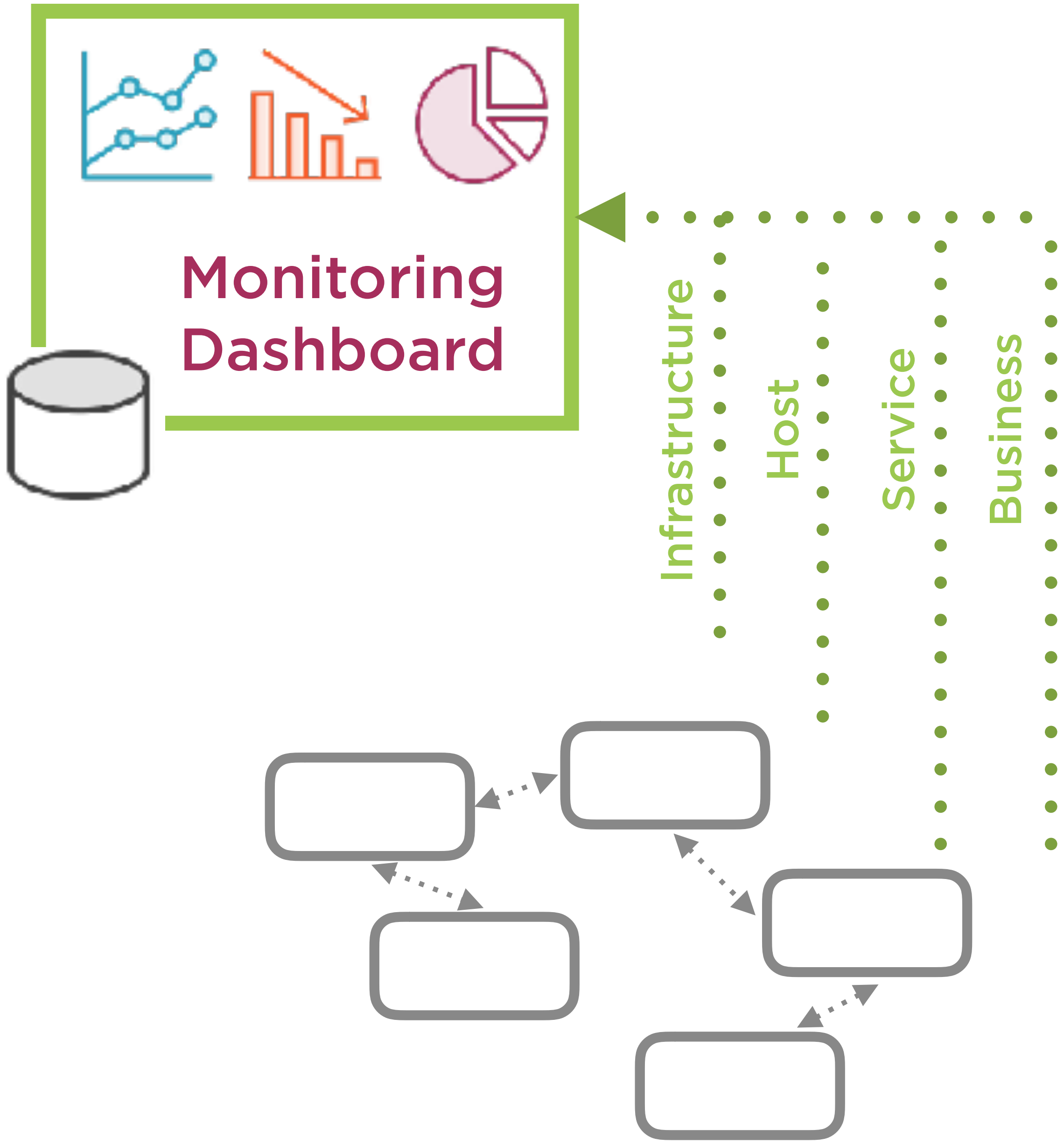
- Monitor dependencies

Use SLA monitoring to alert

- Actionable alerts to fix SLA issues

Communicate SLA issues early

Monitoring Dashboards



Display and collect metrics

Real-time

Centralized, accessible and standardized

Simple, visual and minimal information

- Graphs showing metrics over time

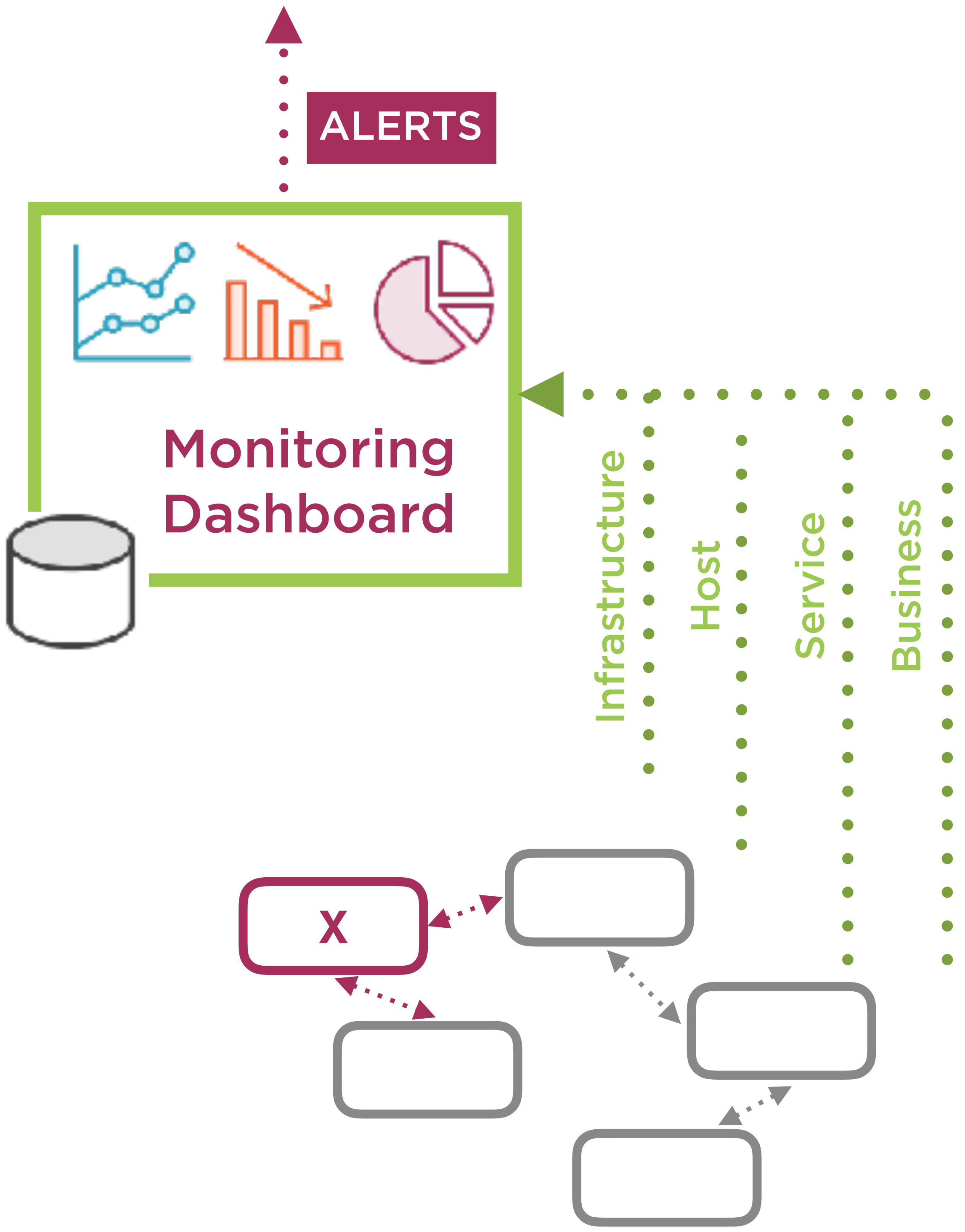
Highlights effectiveness of monitoring

What to display

- Metrics and their alerts
- Metrics for each environment and phase
- Correlate metrics to events

Use to help define thresholds for alerts

Alerting and Monitoring



Real-time monitoring and alerting

When to alert

- Key metrics change
- SLA metrics change
- All metrics surpass thresholds

Make alerts configurable

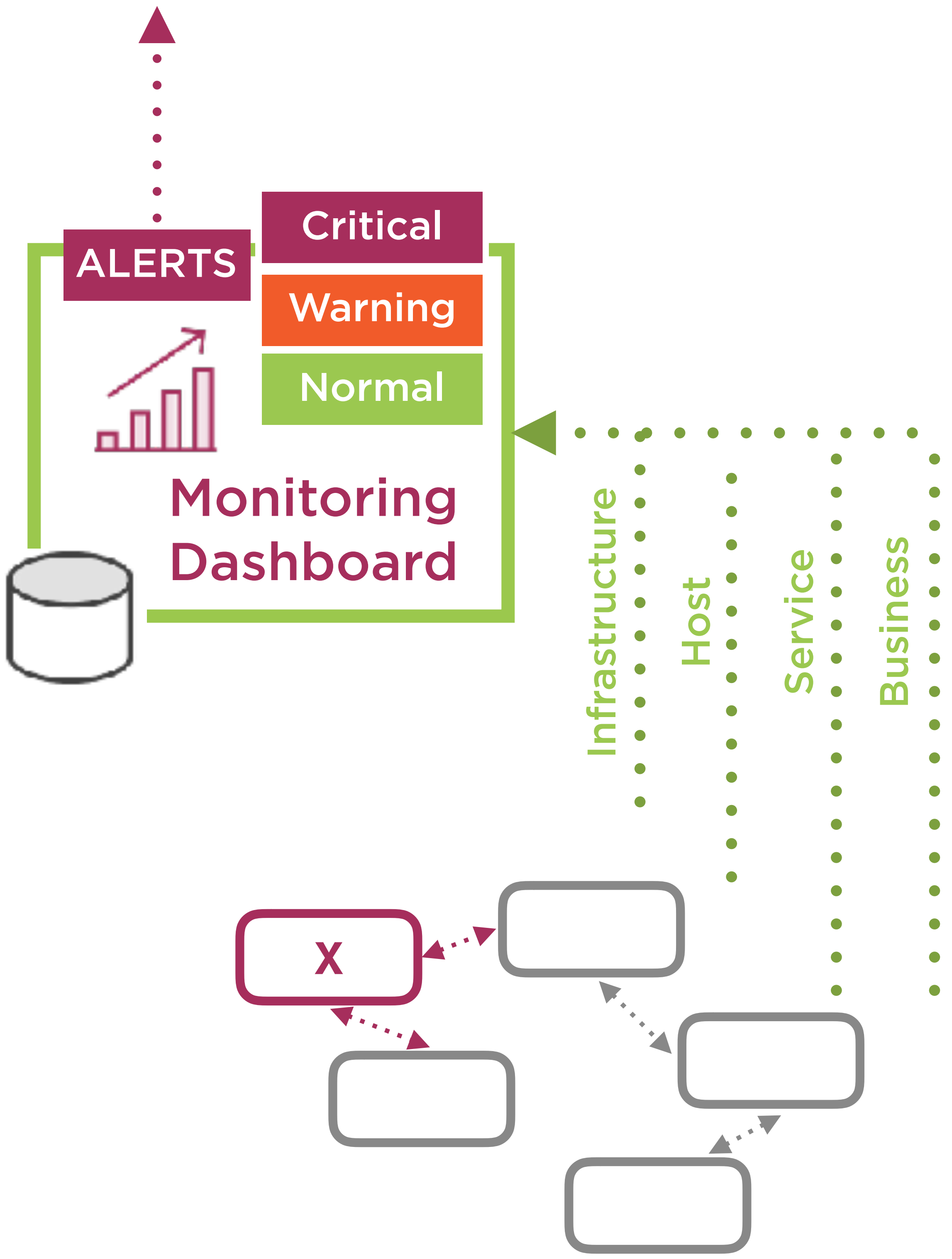
Actionable alerting

Avoid non-actionable alerts

Informative alerts

- Include links to service documentation
- Include links to monitoring dashboard

Defining Thresholds for Alerts



Ranges required for effective alerting

Types of thresholds

- Normal state
- Warning state
- Critical state

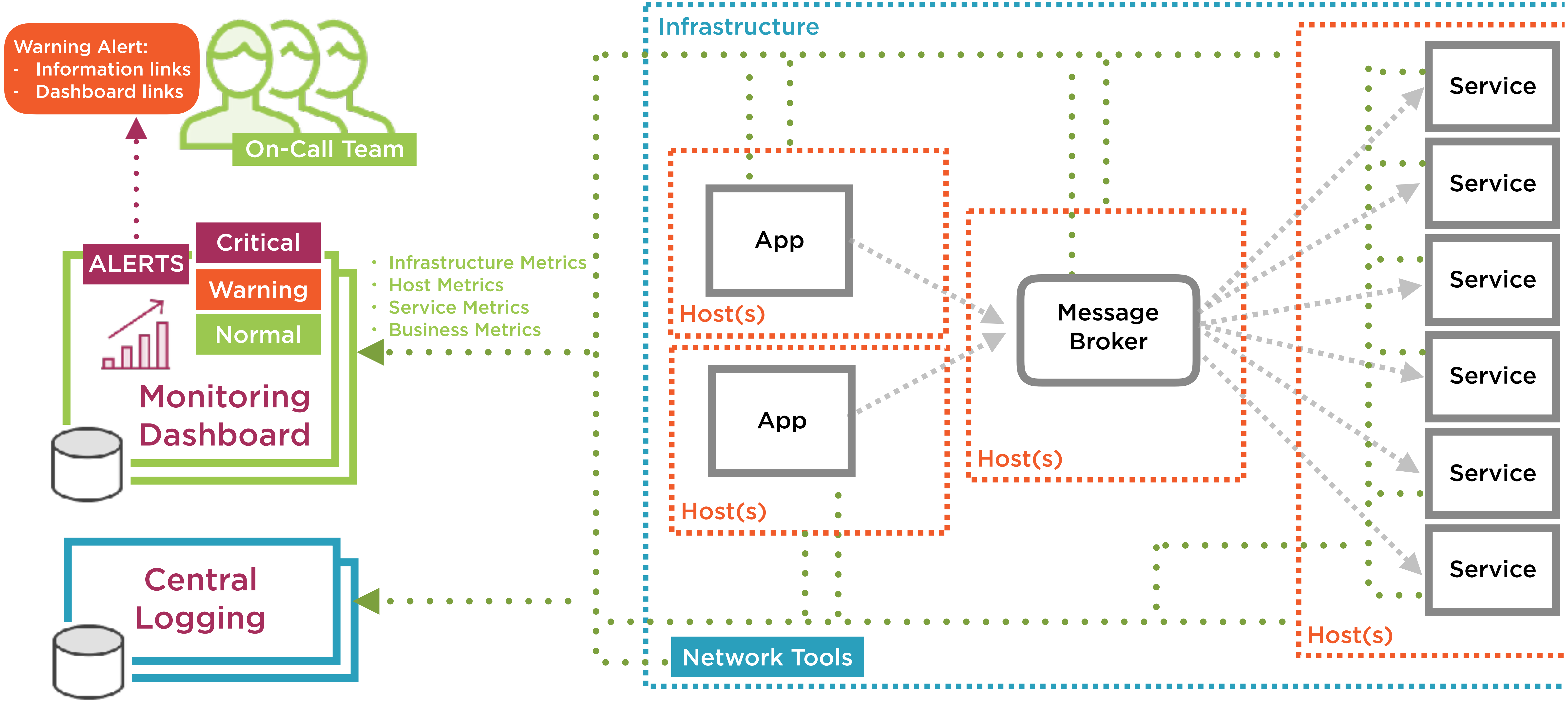
Warning alerts prevent outages

Effective levels for thresholds

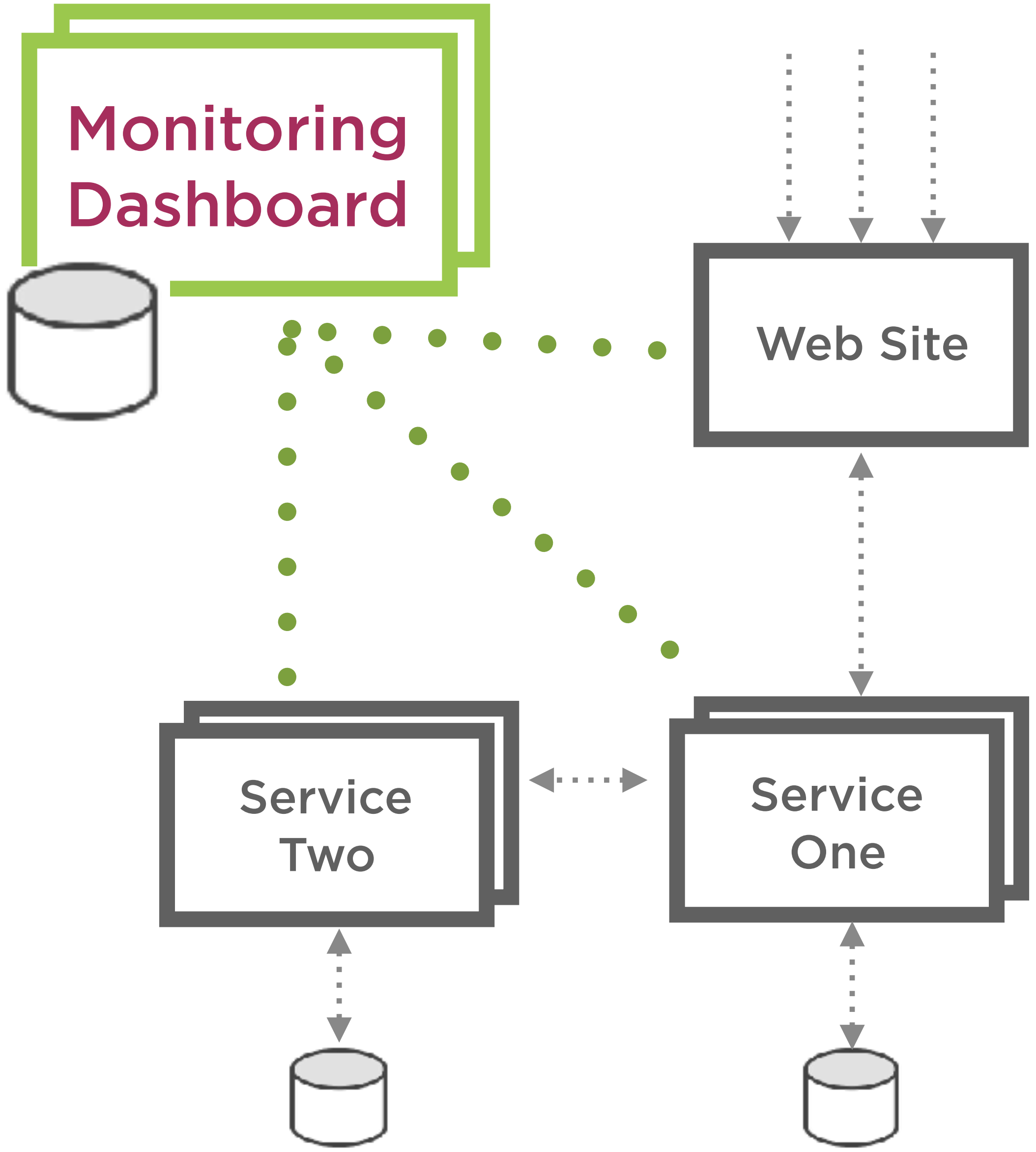
Defining thresholds

- Use historical data
- Performance testing and load testing data
- Expected traffic for normal thresholds
- Larger than expected traffic

Microservices Monitoring Patterns



Monitoring Tools



StatsD

Graphite

New Relic

PRTG

Nagios

Windows performance counters

Bespoke dashboards

Cloud provider tools

- Azure Application Insights

Many more…

Summary

Microservices Monitoring

Monitor Key Metrics

Monitor SLA Metrics

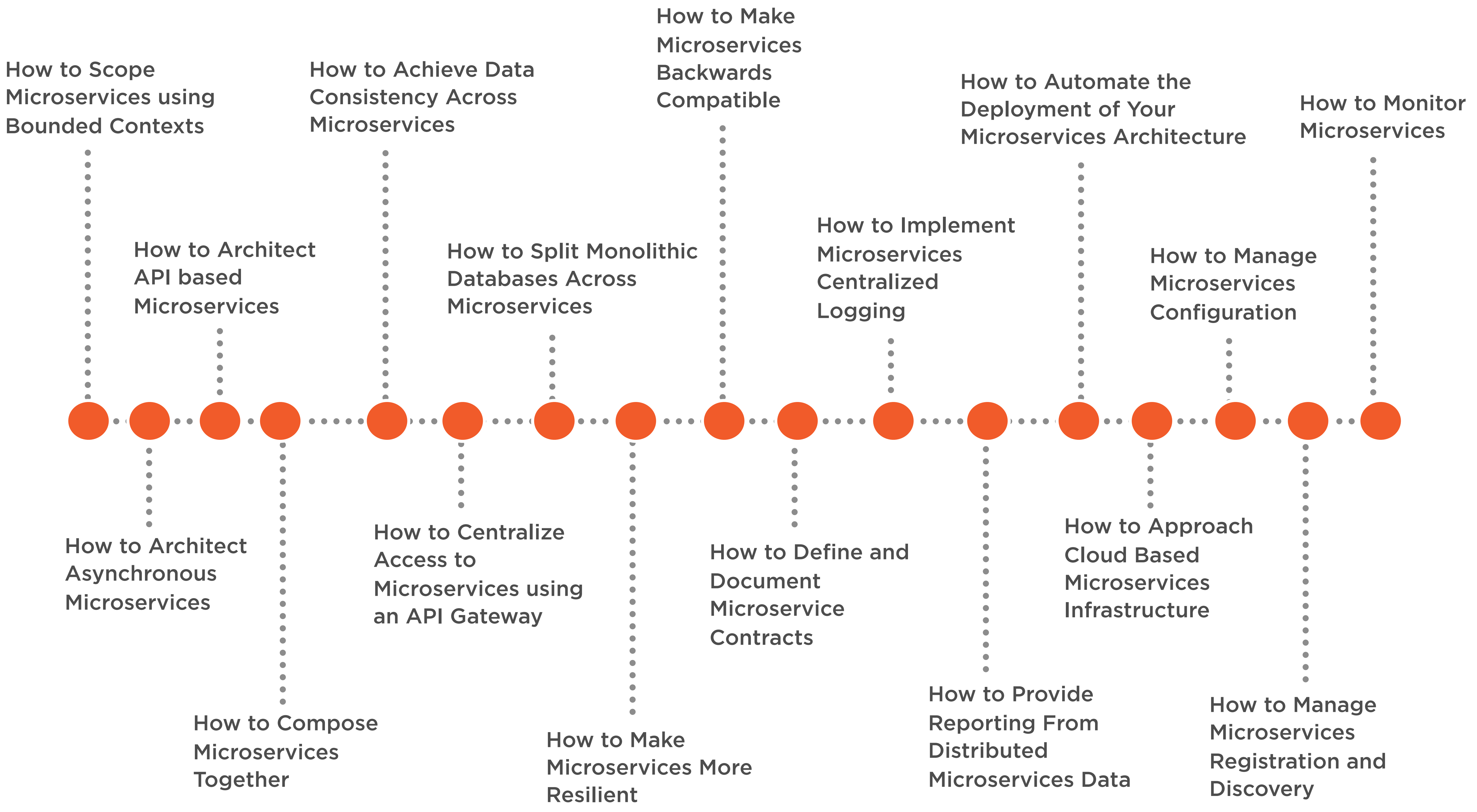
Monitoring Dashboards

Alerting and Monitoring

Defining Thresholds for Alerts

Monitoring Tools and Patterns

Microservices Architectural Design Patterns Playbook



How to Scope Microservices using Bounded Contexts

How to Architect Asynchronous Microservices

How to Architect API based Microservices

How to Compose Microservices Together

How to Achieve Data Consistency Across Microservices

How to Centralize Access to Microservices using an API Gateway

How to Split Monolithic Databases Across Microservices

How to Make Microservices More Resilient

How to Make Microservices Backwards Compatible

How to Define and Document Microservice Contracts

How to Implement Microservices Centralized Logging

How to Provide Reporting From Distributed Microservices Data

How to Automate the Deployment of Your Microservices Architecture

How to Approach Cloud Based Microservices Infrastructure

How to Manage Microservices Configuration

How to Manage Microservices Registration and Discovery

How to Monitor Microservices