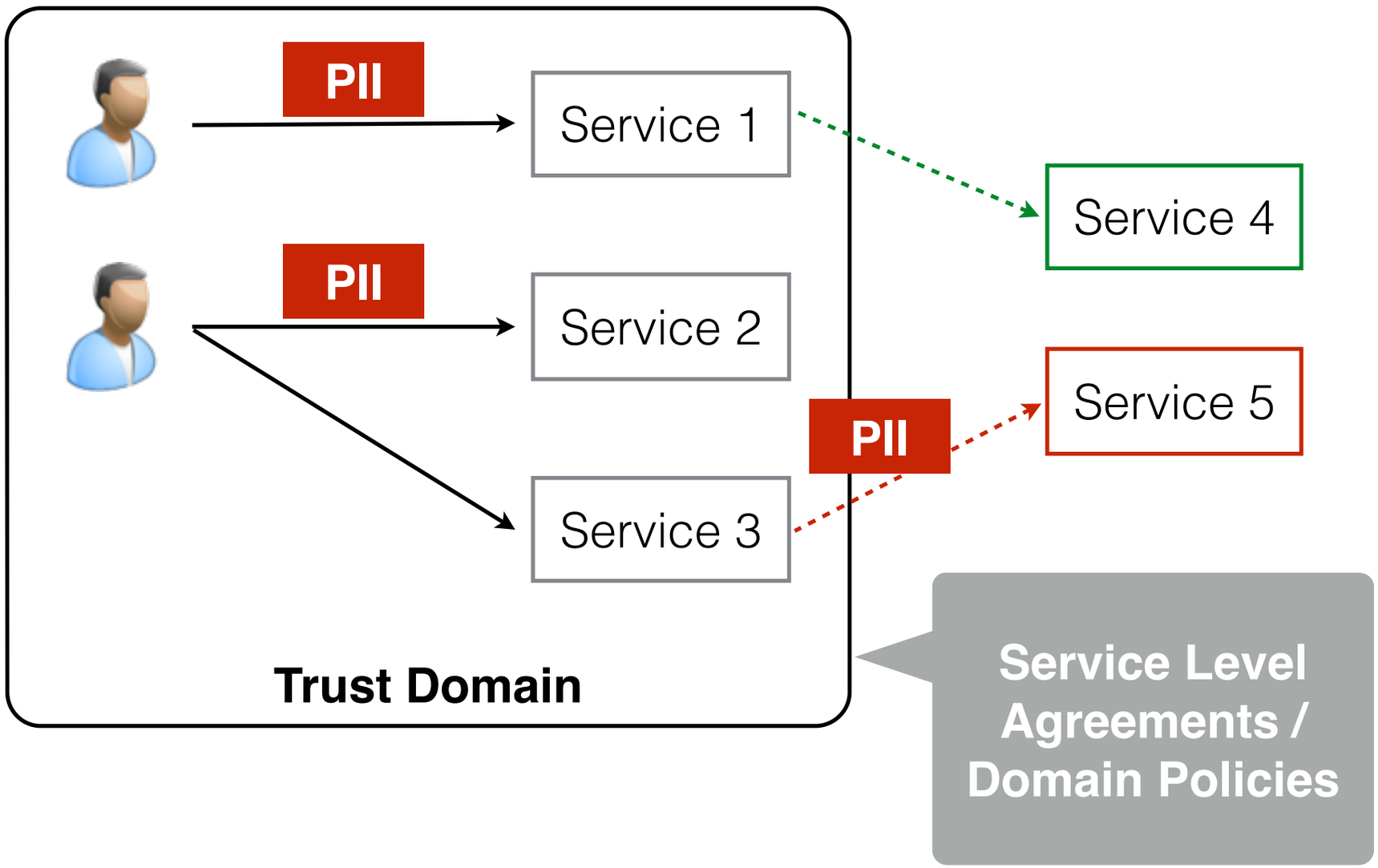
End-to-End Security Policy Auditing and Enforcement in Service Oriented Architecture

Progress Report: January 2014
and
Related Research

Agenda

- Motivation
- REST/SOA Monitoring Framework
- Demo
- Future Work

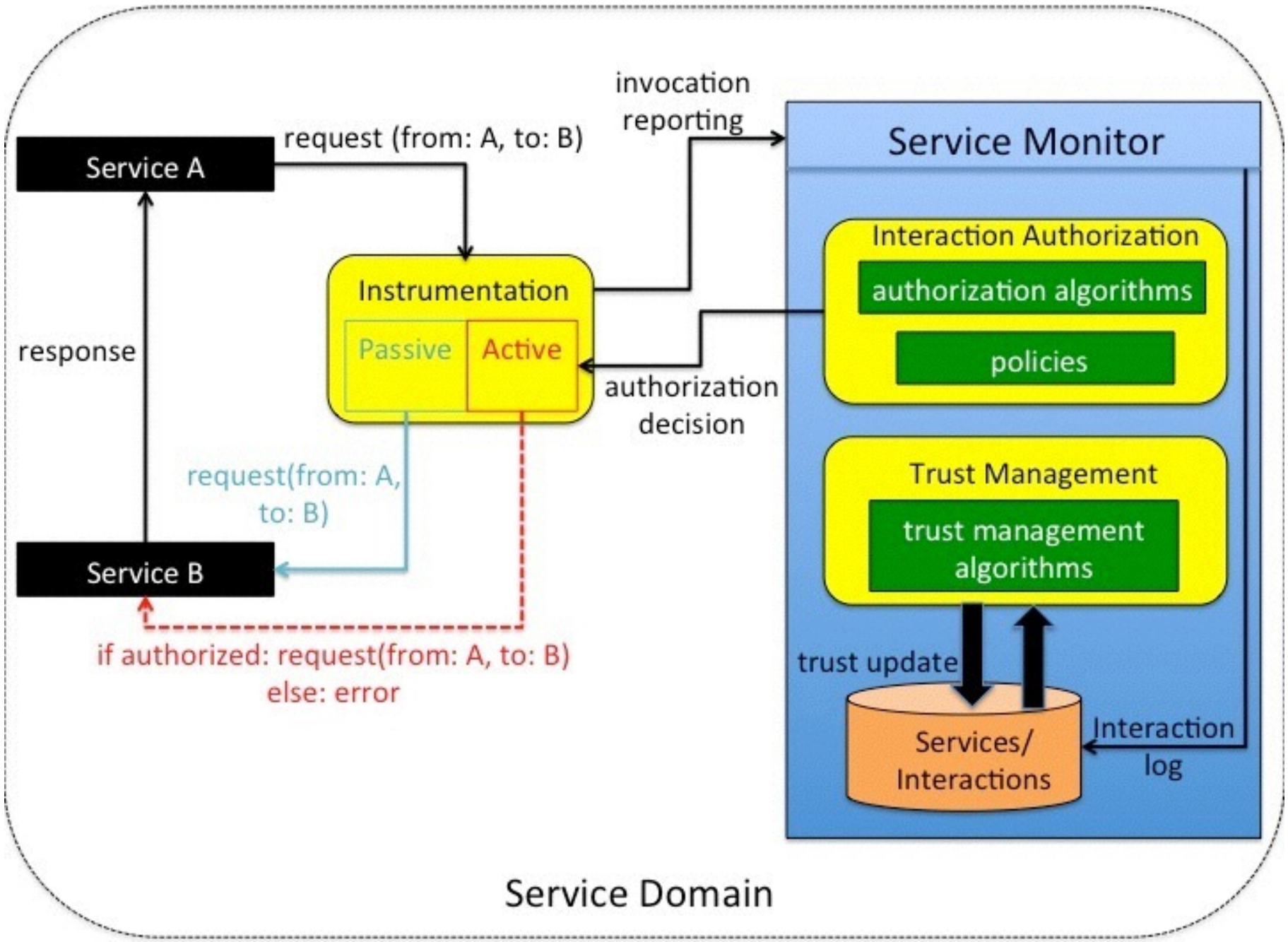
Motivation


PII
Service 1
PII
Service 2
Service 3
Service 4
PII
Service 5
Trust Domain
Service Level Agreements / Domain Policies

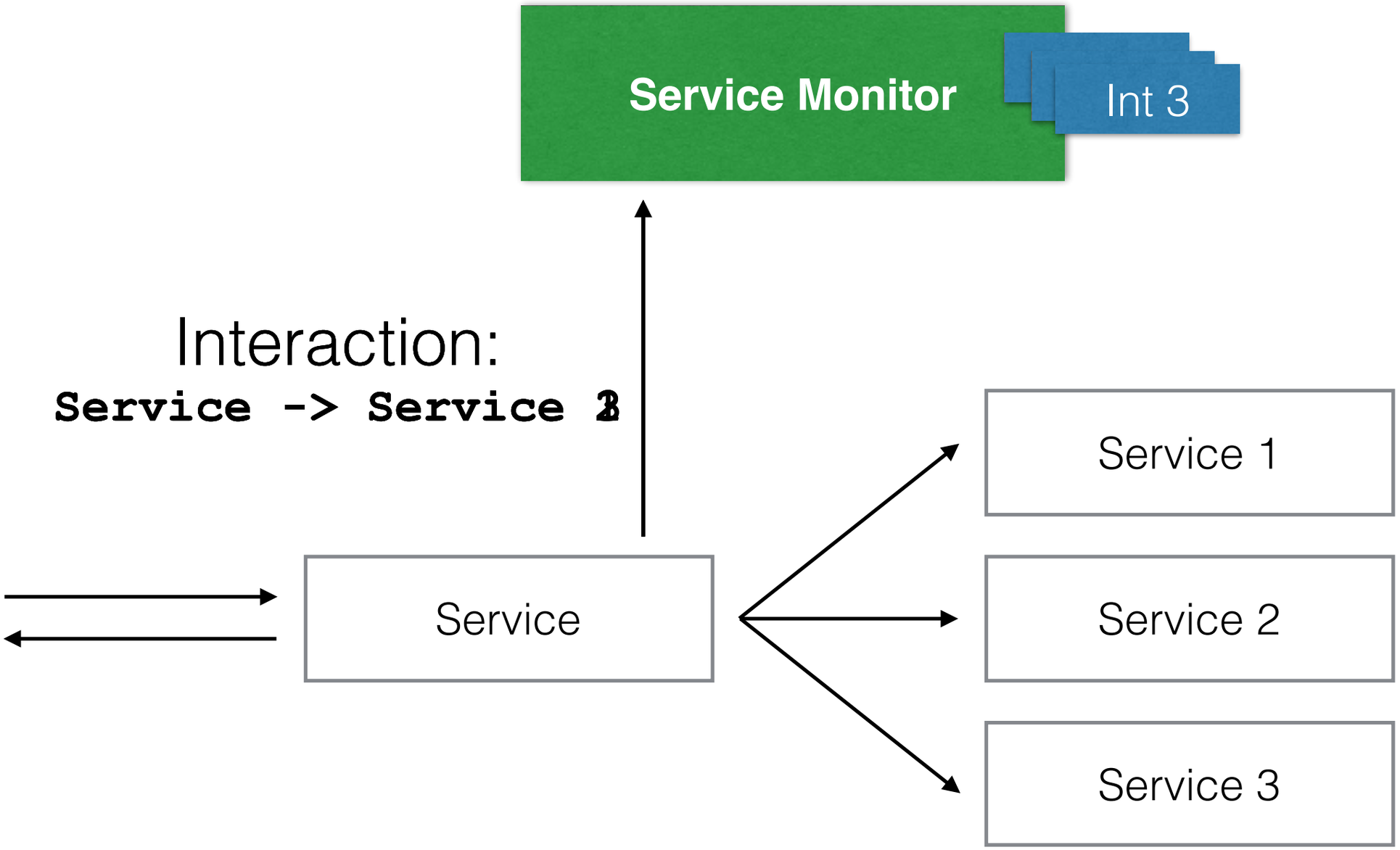
REST/SOA Monitoring

- Remote Monitoring
  - Passive and Active Monitoring
- Service Composition Topologies
- Trust Management
- Service Interaction Authorization

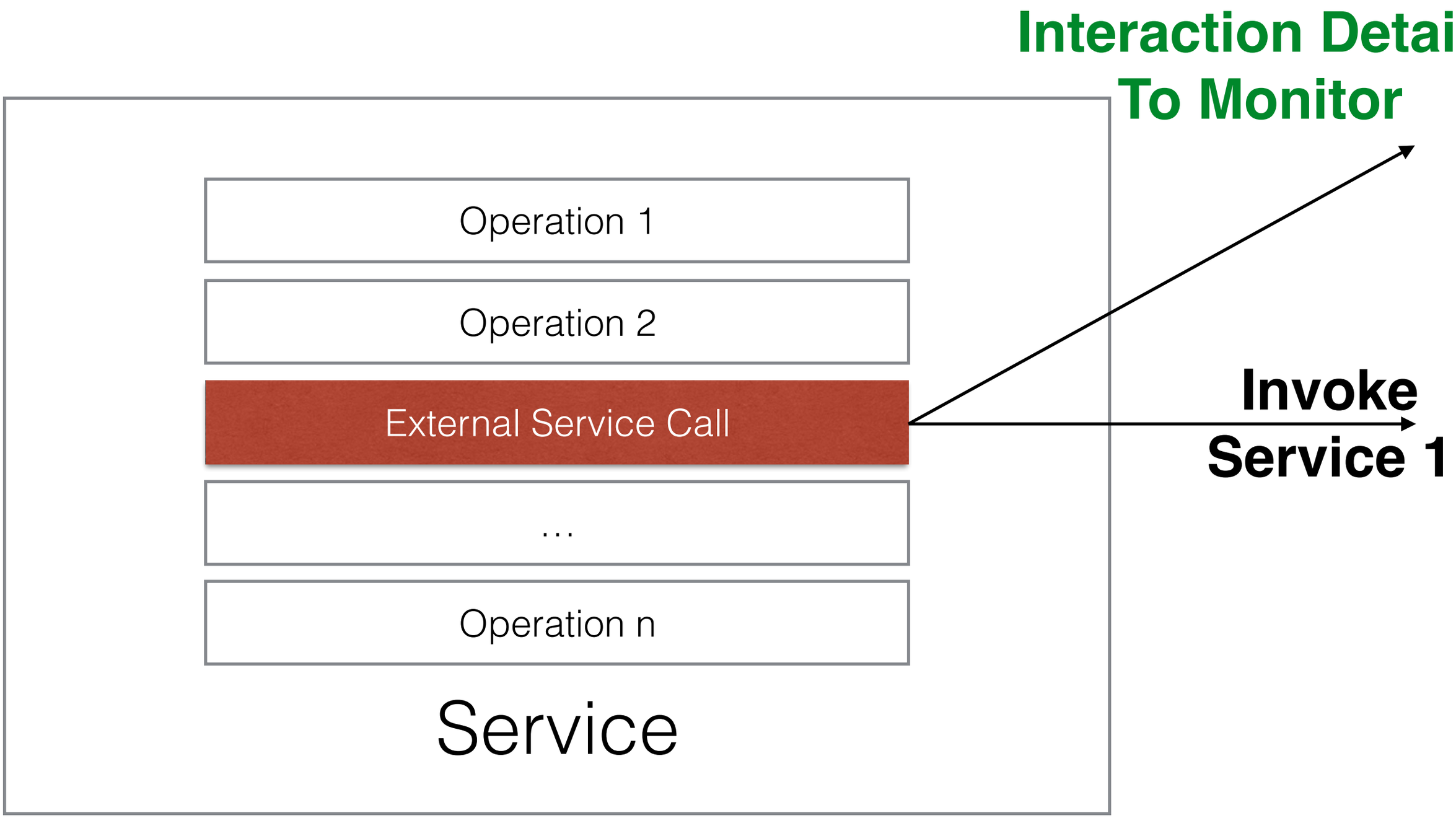
Solution Architecture



Passive Monitoring



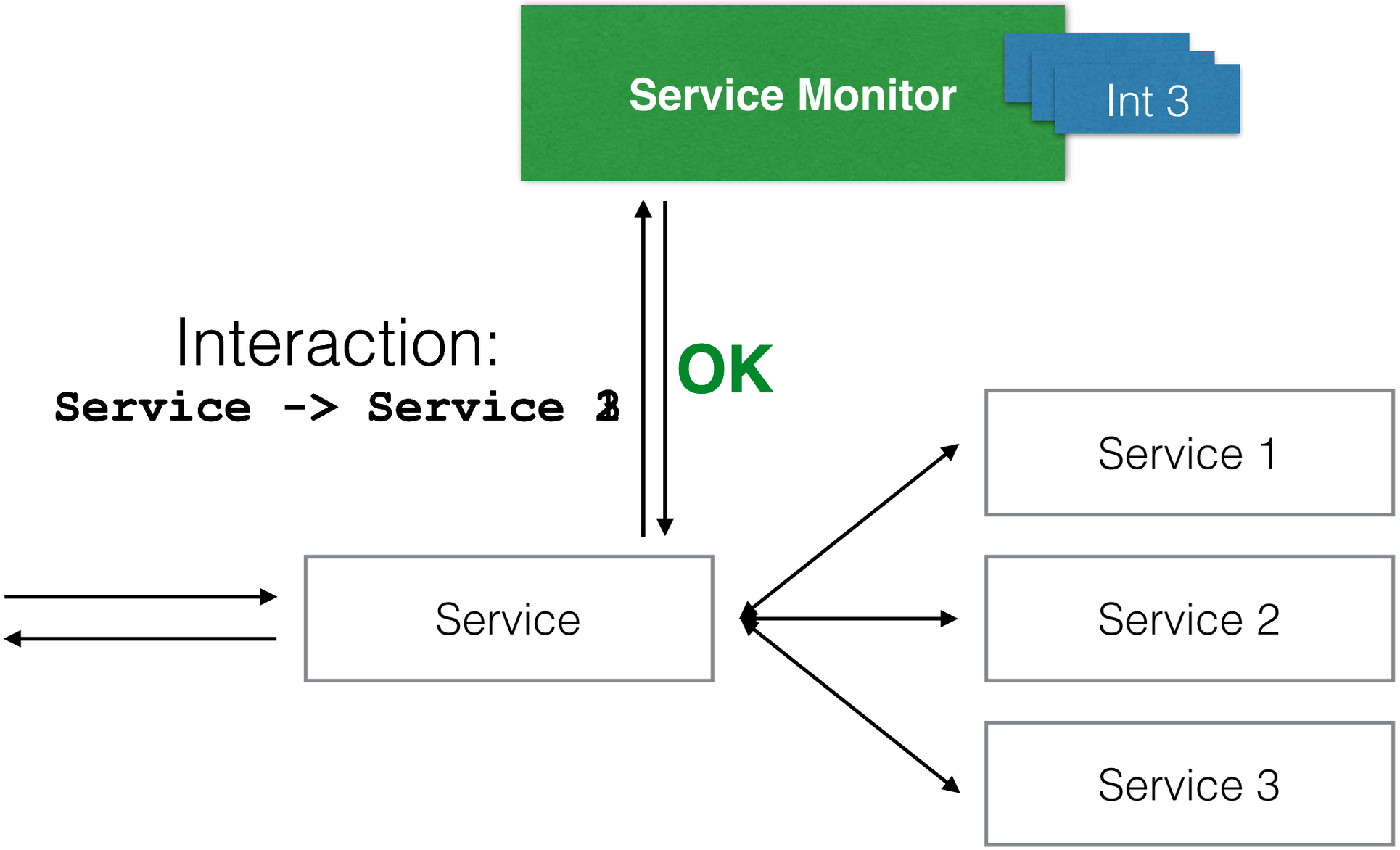
Passive Monitoring



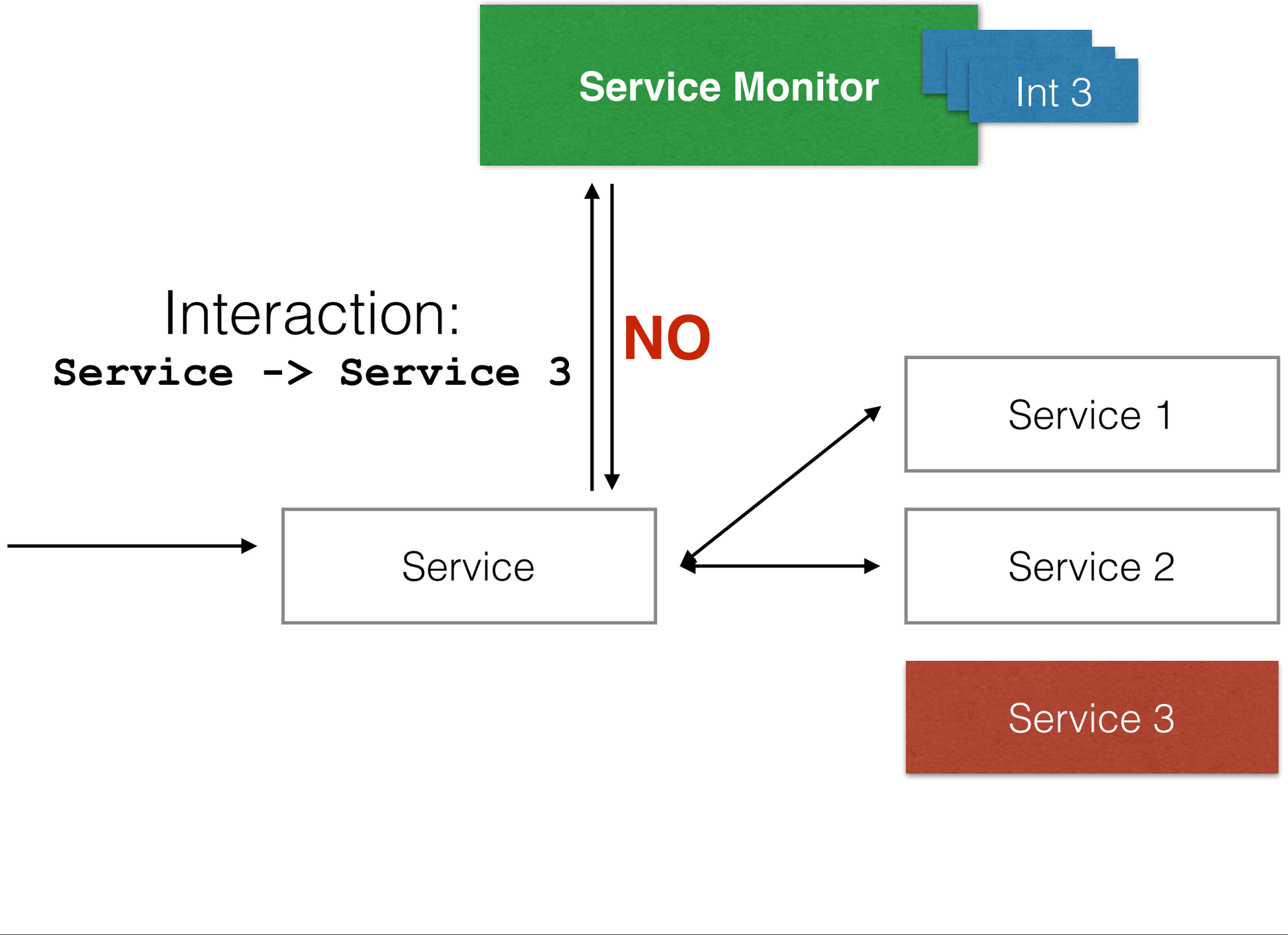
Passive Monitoring

- Service monitor invocation is **transparent** to regular service operation
- Service monitor does not return any information to the monitored service
- Service monitor maintains context information of each service
- Useful for a system administrator to monitor the system in production mode

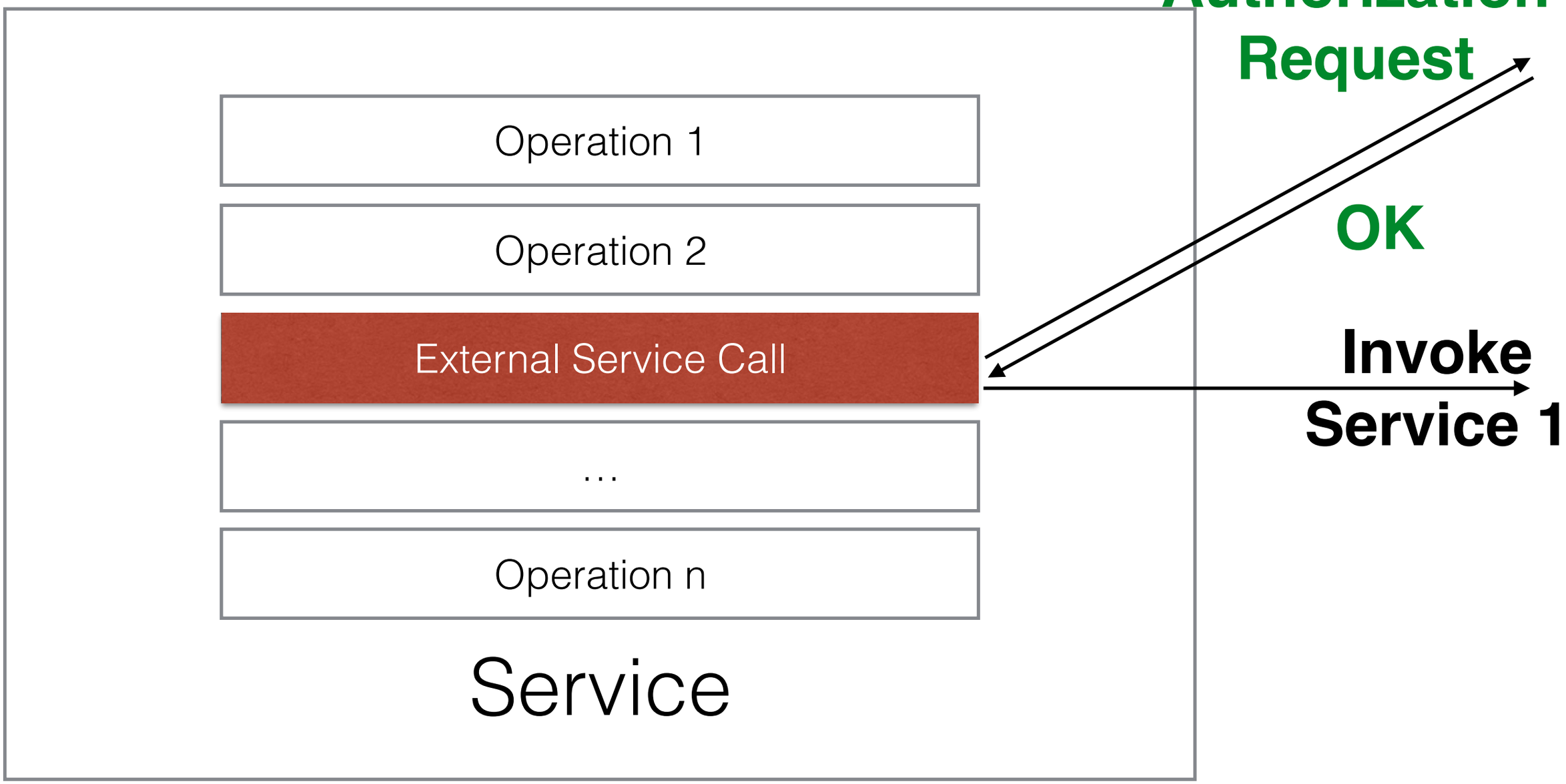
Active Monitoring



Active Monitoring



Active Monitoring



Active Monitoring

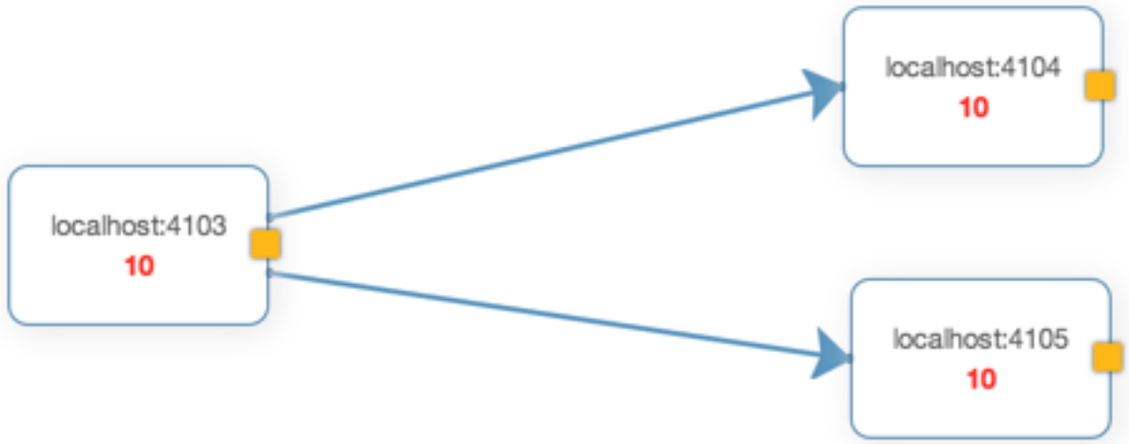
- Service monitor invocation **blocks** regular service operation
- Service monitor returns interaction authorization results
- Decision based on various contextual information such as trust levels, service load, clearance level of invoker etc
- Effective in policy enforcement and to guarantee service level agreements

Service Composition Topologies

- Service Proxy

- Service Chain

- Service Facade



Trust Management

- Trust level is a measure of service behavior over time
- Service level agreements are based on service trust levels
- Service Monitor evaluates service behavior and maintains dynamic trust levels for monitored services
  - Uses service interactions, service level agreements, whitelisting/blacklisting
  - Tracks invocation graphs
  - Propagates changes in trust

Service Interaction Authorization

- Active monitoring requires the service monitor to authorize each service interaction
- Authorization is based on service level agreements and system policies
- Service Monitor makes authorization decisions
- Service instrumentation enforces authorization decisions
  - Controls external service invocations
  - Block, Allow, Redirect etc

Implementation

https://code.google.com/p/end-to-end-soa

Features

- Instrumented service interaction modules
- REST services
- Pluggable service topologies
- Pluggable Trust Management Algorithms
- Pluggable Authorization Algorithms
- Service Monitor Management Console

Instrumented “request”

- **`instr_request`**: Non-blocking instrumentation
  - Sends all invocation metadata to service monitor before and after invocation [asynchronous]
- **`instr_request_block`**: Blocking implementation
  - Waits for authorization form service monitor before allowing interaction
  - When interaction allowed carries out interaction and sends interaction metadata after the interaction [asynchronous]

REST Services

- Services implemented as node.js/express applications
- Registered with the Service Monitor
- Exposes a REST API to be consumed by other services
- Message format: JSON
- Allowed operations: GET, PUT, POST, DELETE

Topology Implementation

- `instr_request` or `instr_request_block` used with the typical request syntax when interacting with external services
- Scenario definition added into `passive` or active directory in `monitor/scenarios/`
- Graphical representation of the scenario
- Management Console:
  - Ability to update trust levels
  - Scenario invocation

Pluggable Trust Algorithms

- Each algorithm is a self-contained module

```
module.exports = {name:'My First Algo', alg:my_algo};
```

- Loaded by service monitor on bootstrap
- Ability to enable/disable a trust management algorithm
  - Simple Average
  - Moving Average
  - Lower Trusted Service Access Denied

Pluggable Authorization Algorithms

- Each algorithm is a self-contained module

```
module.exports = {name:'Authz Algo', authorize : simple_auth};
```

- Authorization decision is carried out by the `authorize(from, to)` function of the module

- Ability to enable multiple authorization algorithms

Active Interaction Authorization Algorithms

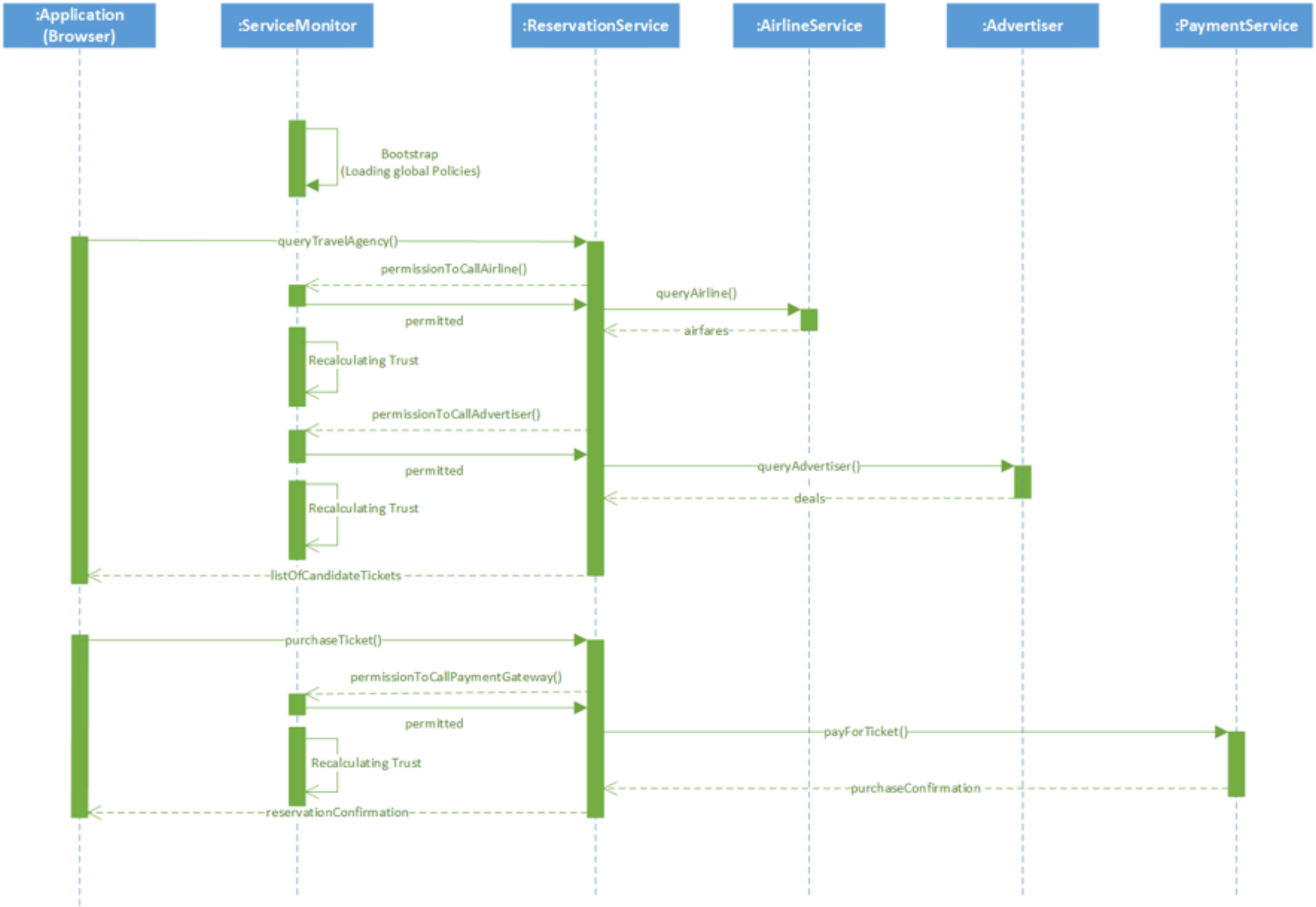
- Simple trust level based authorization
- XACML based interaction authorization
    - Resource: Target service
    - User: Service invoking the target service
    - Action: READ/WRITE
    - Environment
        - Conditions on which access is allowed or denied
- Based on WSO2 Balana XACML Implementation

XACML Environment

- Trust levels of the service and target
  - Eg: Prevent access of services with trust level < 5
- Certain times of day the access is not allowed
  - Eg: Prevent access of a service from 1100 to 1200
- Certain load levels at which access is not allowed
  - Eg: Load threshold = 50%

DEMO

Travel Reservation Service Scenario

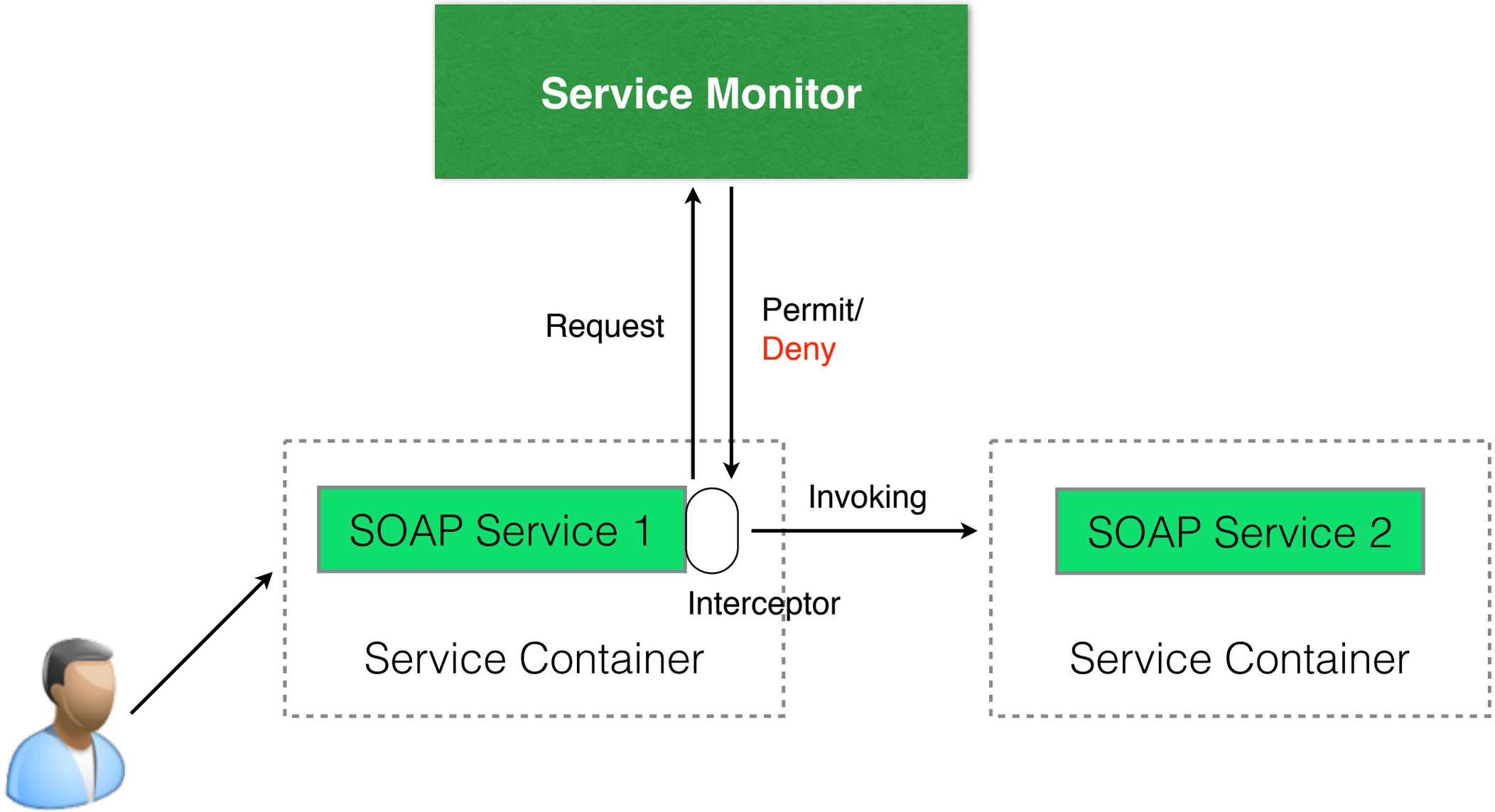


Future Work

SOAP Service Monitoring

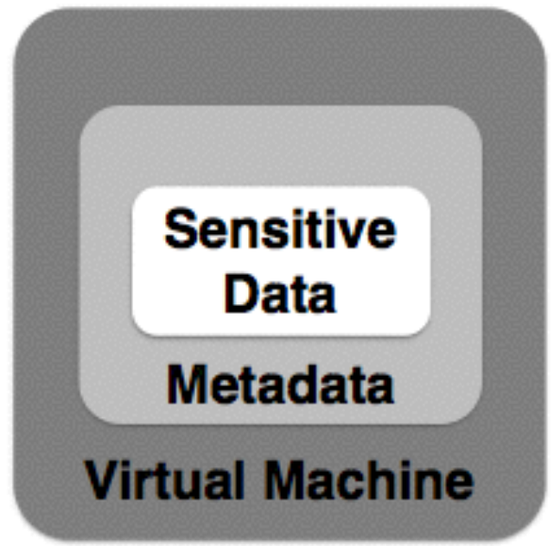
- Services will be implemented in Apache Axis2 Server
- A module to intercept the SOAP messages using an Axis2 handler
- The interceptor will send a request to Service Monitor to validate it based on system policies.
  - Active and Passive Monitoring Modes
  - Aspect Oriented Programming (AOP) based instrumentation
- Set of common service composition scenarios

SOAP Service Monitoring

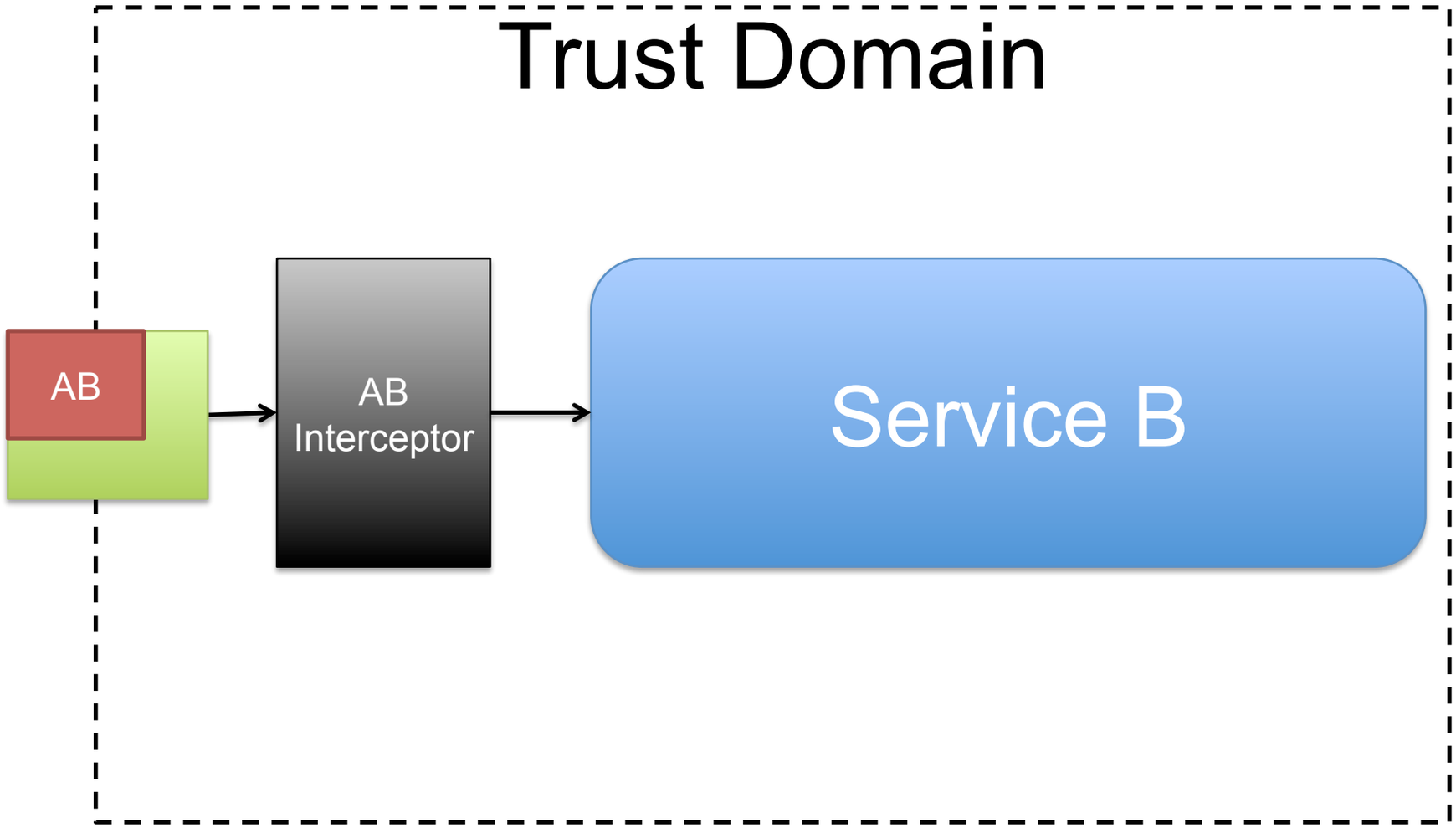


Active Bundles in SOA

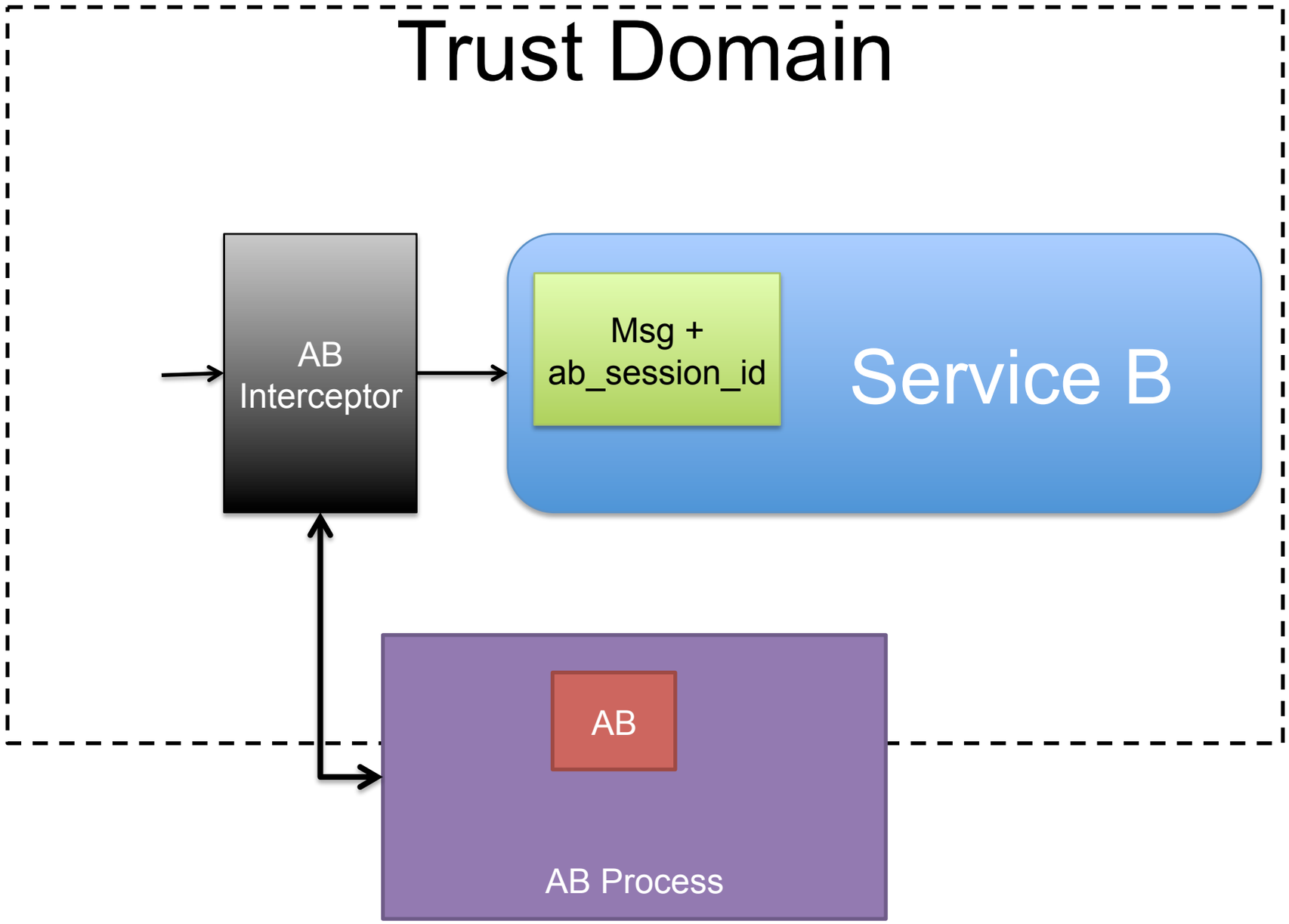
- Active Bundle (AB) is a data protection mechanism
- AB exposes an API to services
  - `getSLA()`
  - `authenticateChallenge()`
  - `authenticateResponse(token, signedToken, serviceCert)`
  - `getValue(sessionKey, dataKey)`
- AB API implemented using Apache Thrift
- AB is included in the SOAP header



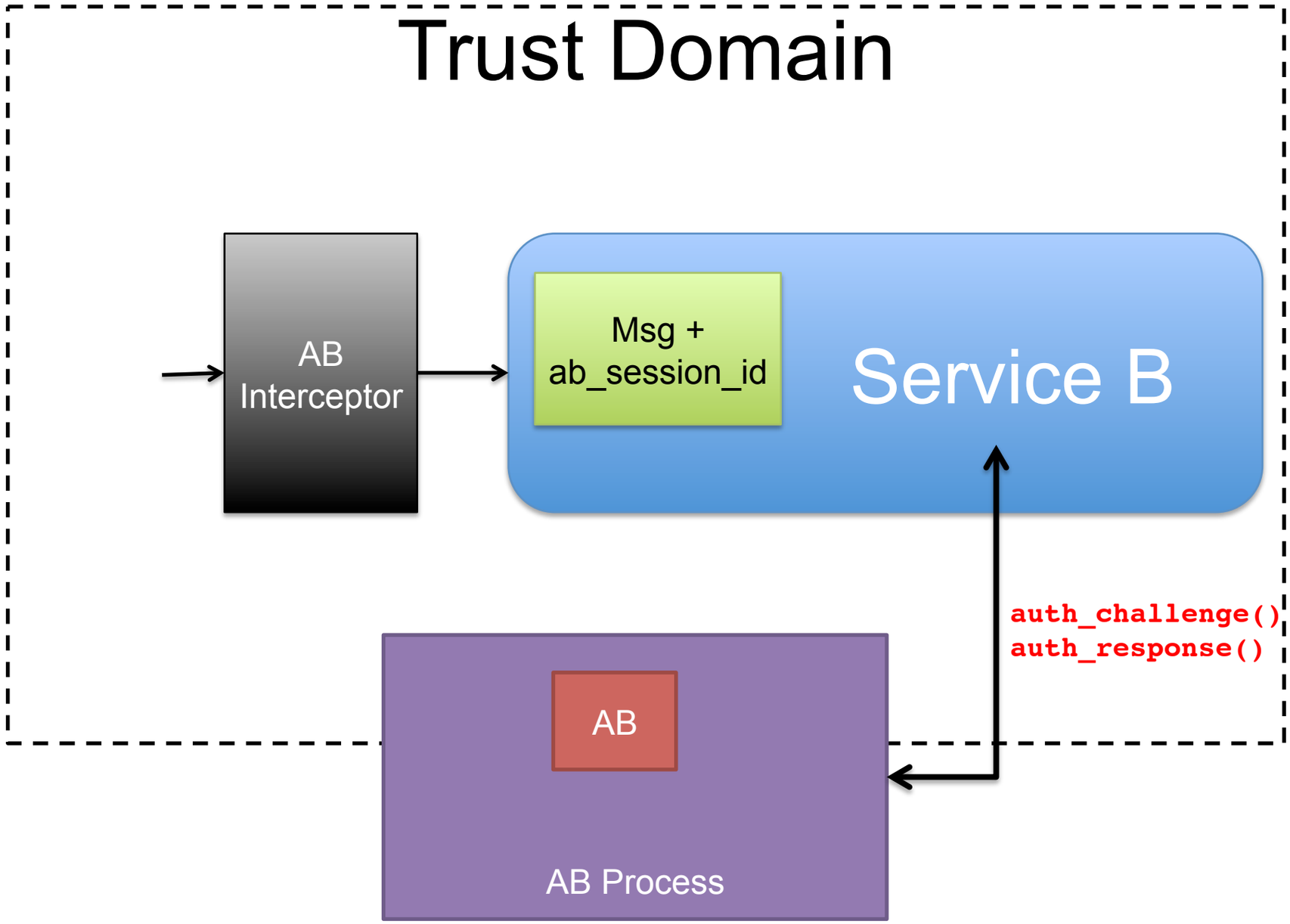
AB-Service Interaction



AB-Service Interaction



AB-Service Interaction



AB-SOA Implementation

- Services are deployed on Apache Axis2
- AB Interceptor
  - Implemented as an Apache Axis2 module
  - Extracts AB from the message
  - Authenticates and verifies integrity of the AB
  - Executes AB as an independent process
  - Adds AB process information into the message context