Service Management Interface

for Service Developers

Agenda

- Introduction
- ManagementReport
- Use Cases
- Integration Guidelines
- Takeaways

Agenda

- Introduction
- ManagementReport
- Use Cases
- Integration Guidelines
- Takeaways

Service Management Interface Capabilities

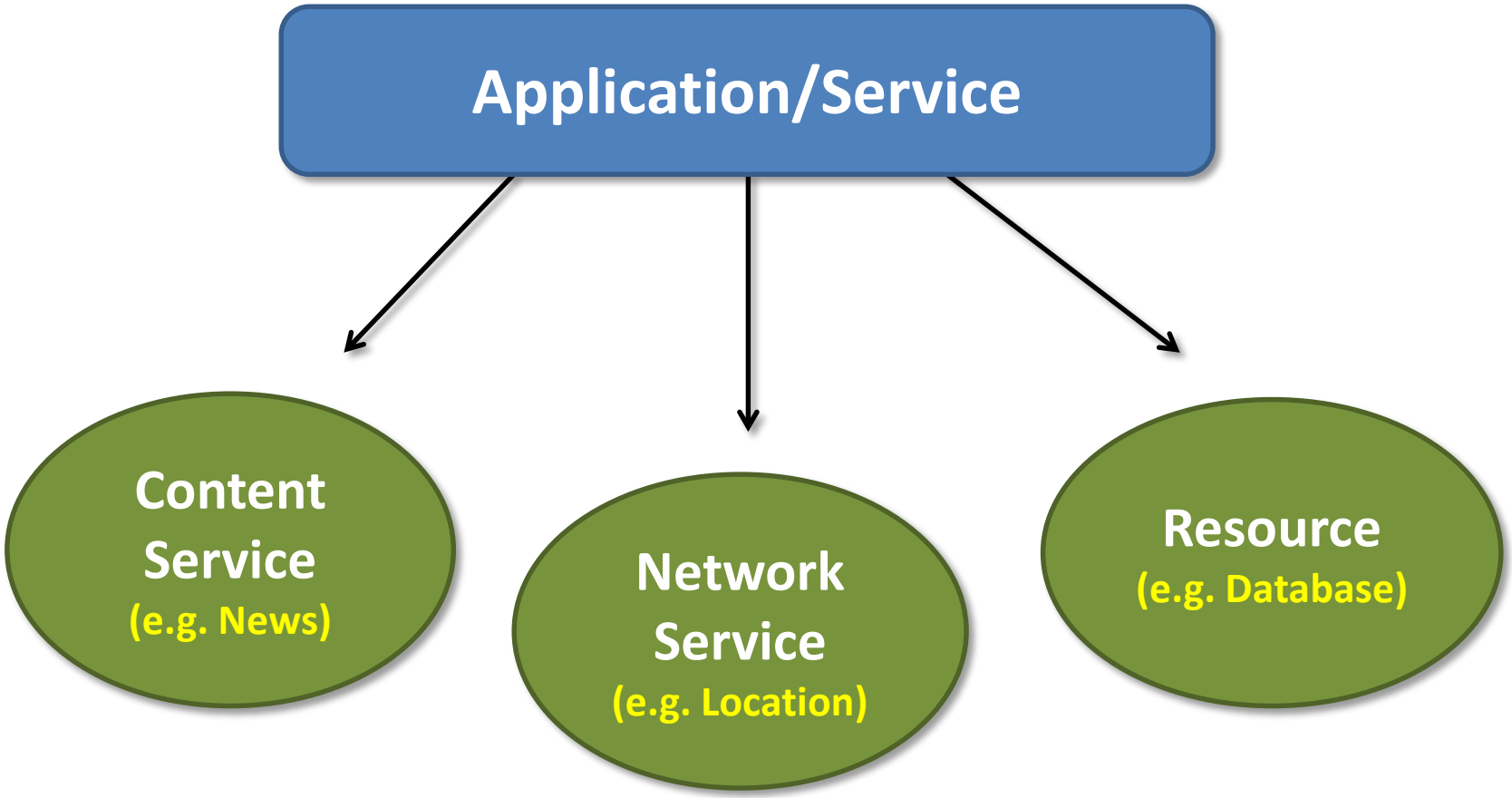
as in Service Delivery Framework Reference Architecture (TMF061)

- **Activation**
  - Making the Service available for a particular context;
- **Provisioning**
  - Configuring the settings of a Service or a Service instance;
- **State Monitoring**
  - Querying the history and current status in terms of life cycle management [for a specific instance of the Service] and our listening for status updates;
- **Usage Monitoring**
  - Querying for usage metrics from the Service instance or listening for usage metrics reports or alarm (e.g. if metrics conditions imply notifications);
- **Health Monitoring**
  - Querying for health metrics from the Service instance or listening to alarm from the resource;
- **Update Configuration**
  - Modification of the setting or life cycle management status of a Service instance;
- **De-activation**
  - Making the Service unavailable in a particular context.

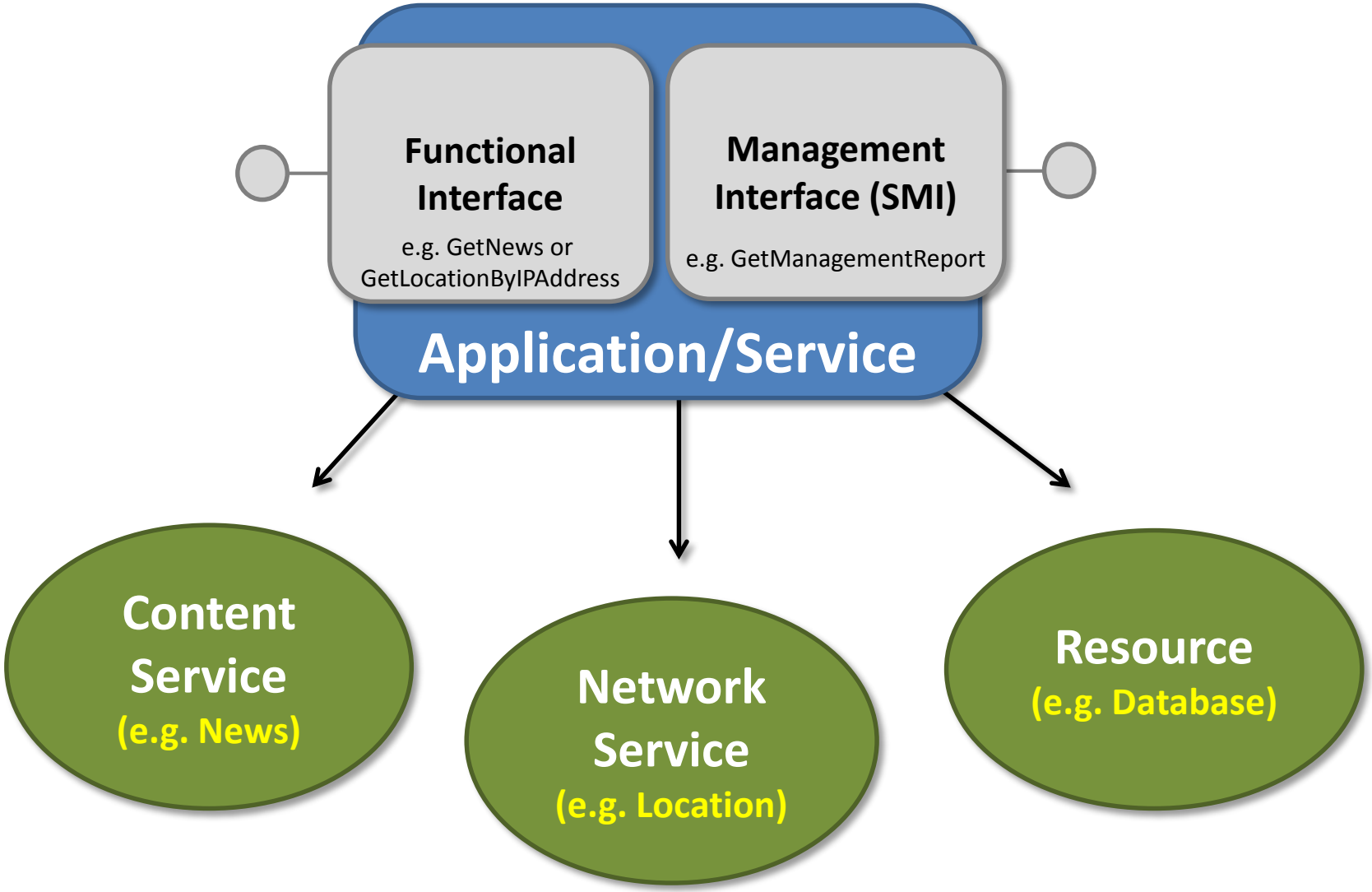
Service Management Interface

- Get/Set ExecutionState
- Get/Set ServiceConfiguration
- Get ManagementReport
- SetNotificationAddress

Application/Service Dependencies



Application/Service Interfaces



Application/Service Consumers

Application Users

Functional Interface

e.g. GetNews or GetLocationByIPAddress

Management Interface (SMI)

e.g. GetManagementReport

Application/Service

Monitoring and Reporting

Content Service (e.g. News)

Network Service (e.g. Location)

Resource (e.g. Database)

Agenda

- Introduction
- ManagementReport
- Use Cases
- Integration Guidelines
- Takeaways

Service Management Interface

ManagementReport Example (1/2)

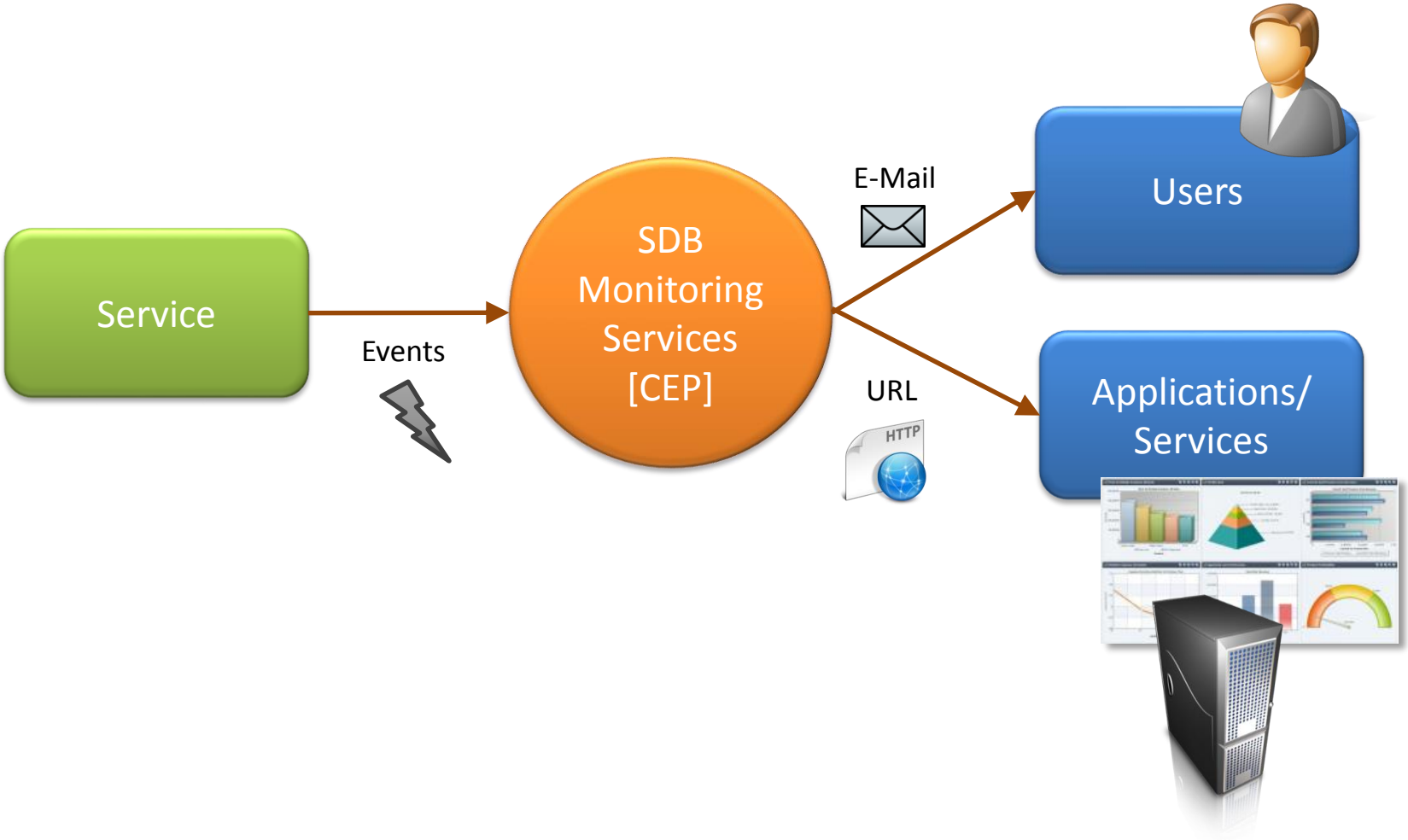
```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
   <GetManagementReportResponse xmlns="http://schemas.tmforum.org/2011/08/ses/smi">
      <GetManagementReportResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <ID>91c334aa-1552-4700-9fd8-198bc73e4094</ID>
        <SourceID>48FCCA1D-8F84-47A0-BFDA-80B8819F87D0</SourceID>
        <DateTime>2013-07-03T17:42:04.3696339+01:00</DateTime>
        <State>
          <Execution>Active</Execution>
          <Health>Operational</Health>
          <Failures/>
        </State>
      </GetManagementReportResult>
    </GetManagementReportResponse>
  </s:Body>
</s:Envelope>
```

Service Management Interface

ManagementReport Example (2/2)

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetManagementReportResponse xmlns="http://schemas.tmforum.org/2011/08/ses/smi">
      <GetManagementReportResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <ID>f35ea993-0b54-490e-9dd2-a4d014750bea</ID>
        <SourceID>48FCCA1D-8F84-47A0-BFDA-80B8819F87D0</SourceID>
        <DateTime>2013-07-03T18:01:17.8649857+01:00</DateTime>
        <State>
          <Execution>Active</Execution>
          <Health>OperationalWithFailures</Health>
          <Failures>
            <Failure>
              <FailureID>DB.0001</FailureID>
              <SourceID>A3E19321-CC49-4BA5-95A6-5FF37AA50FD5</SourceID>
              <Detail>
                Unable to connect to database.
                Exception Message: Failed to establish a connection to '10.135.65.223'.
                Exception StackTrace:    at Npgsql.NpgsqlClosedState.Open(NpgsqlConnector context) in
C:\projects\Npgsql2\src\Npgsql\NpgsqlClosedState.cs:line 204
              </Detail>
            </Failure>
          </Failures>
        </State>
      </GetManagementReportResult>
    </GetManagementReportResponse>
  </s:Body>
</s:Envelope>
```

Event-Driven Notifications



Notification Example (1/2)

ManagementReport (JSON)

```
Payload=
{
      "ID":"fa8cc7e2-e4f3-416b-b5fb-6d91e65d19ed",
      "SourceID":"6c8b9248-6323-4400-8bd9-69b18025f467",
      "DateTime":"2014-05-12T17:35:01.3104577Z",
      "State":
      {
            "Health":"OperationalWithFailures",
            "Execution":"Active",
            "Failures":
            [
                  {
                        "FailureID":"a872a649-5eca-4735-8a1e-0f5cd94cda22",
                        "SourceID":"6c8b9248-6323-4400-8bd9-69b18025f467",
                        "Details":
                        {
                              "IPAddress":"176..78.121.243",
                              "failureMessage":"Invalid IP Argument",
                              "clientIPAddress":"",
                              "EsbPrimaryId":""
                        }
                  }
            ]
      }
}
```

Notification Example (2/2)

ManagementReport (eMail)

seg 12/05/2014 18:35

do-not-reply@services.sapo.pt

SAPO GIS

To João Filipe Rosado Gouveia

Management Report

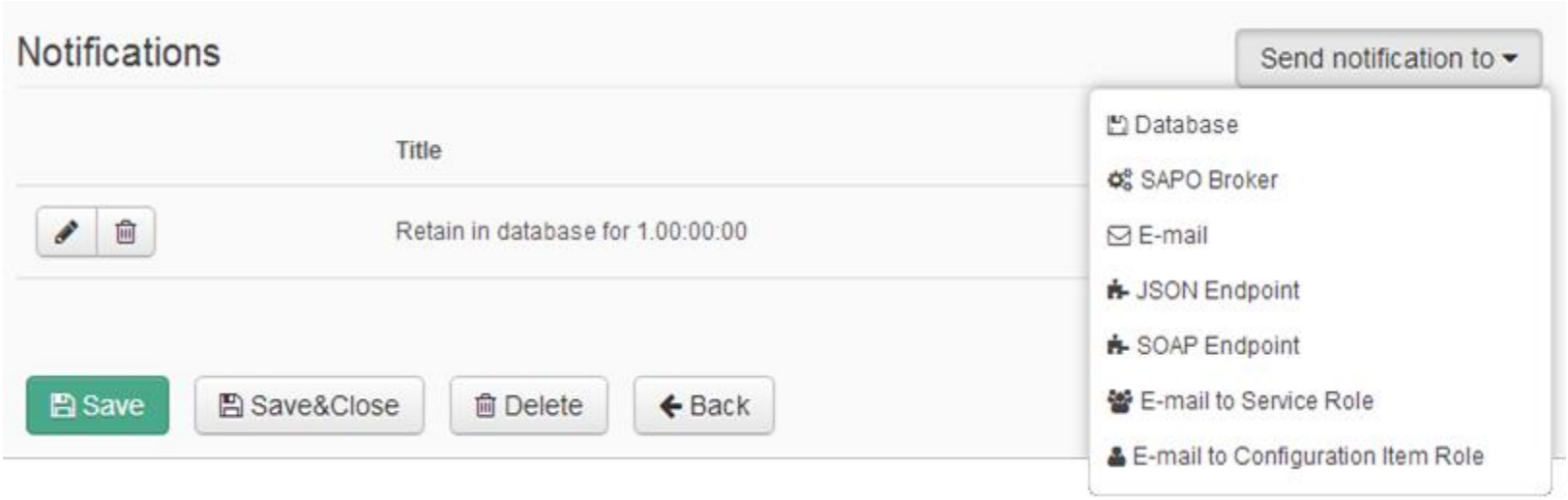
Report ID:	fa8cc7e2-e4f3-416b-b5fb-6d91e65d19ed
Publish Date:	2014-05-12T17:35:01.3100000Z
Source:	**GIS/GetLocationByIPAddressWithNetworkDataCLF***Web Service Operation* 6c8b9248-6323-4400-8bd9-69b18025f467
Activity ID:	
Health State:	**OperationalWithFailures**
Execution State:	Active

Failures

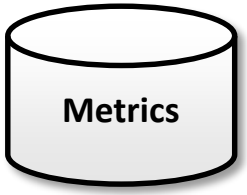
a872a649-5aca-4735-8a1a-0f3cd94cda22 — Source: **GIS/GetLocationByIPAddressWithNetworkDataCLF***Web Service Operation* 6c8b9248-6323-4400-8bd9-69b18025f467

```
<Details>
  <IPAddress>176..78.121.243</IPAddress>
  <failureMessage>Invalid IP Argument</failureMessage>
  <clientIPAddress>
  </clientIPAddress>
  <EsbPrimaryId>
  </EsbPrimaryId>
</Details>
```

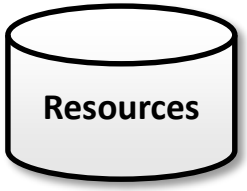
Notifications Management



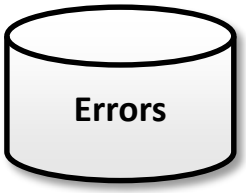
Service Management Interface Data Sources



- **Metrics Catalogue**
  - Common metric codes and descriptions
    - E.g. Metric “123” is “Requests Per Second”



- **Service/Resource Dependencies**
  - Graph of ascendant/descendant dependencies codes and descriptions
    - E.g. Application “A1” depends on Service “S1” and Service “S1” depends on database “DB1”



- **Reference Data**
  - Common error codes and messages
    - E.g. Exception “789” is a “Database connection error”

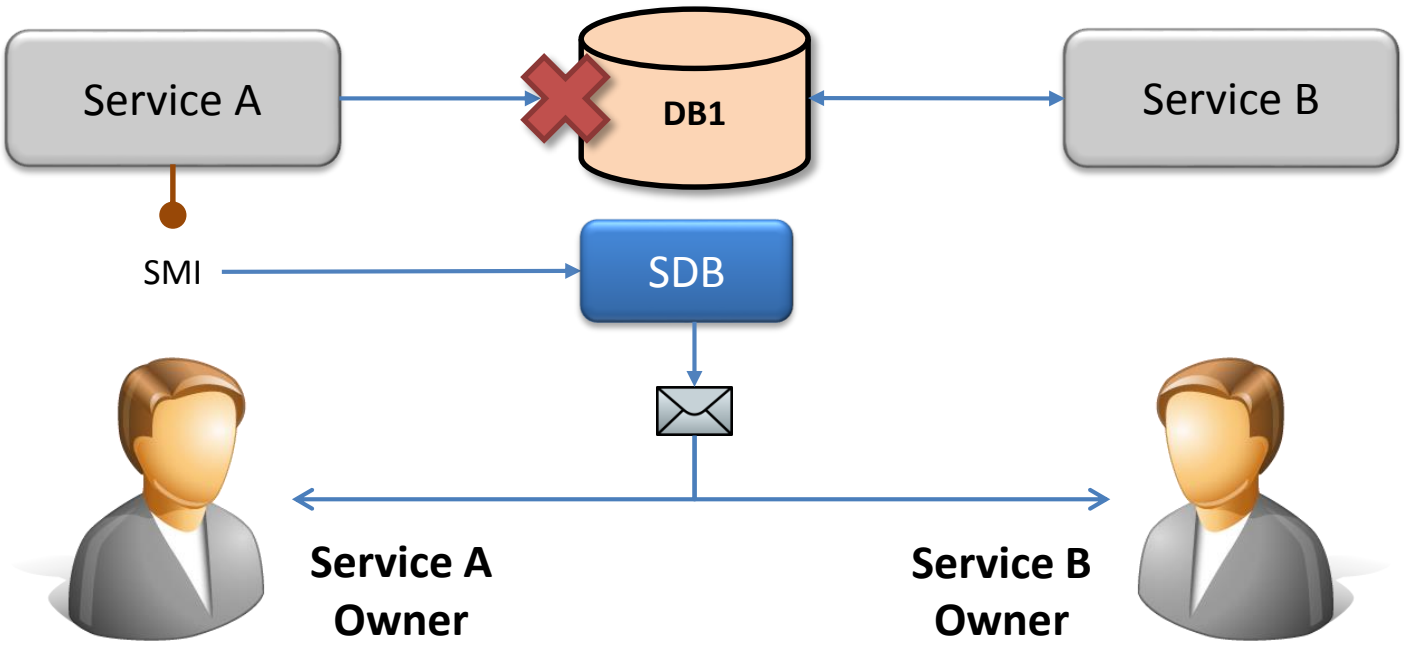
Agenda

- Introduction
- ManagementReport
- Use Cases
- Integration Guidelines
- Takeaways

Service Management Interface

Use Cases (1/2)

- Service A reports that its database dependency DB1 is not available.
- Because Service B also uses the same database, both Service A and B owners receive a service dependency failure notification.



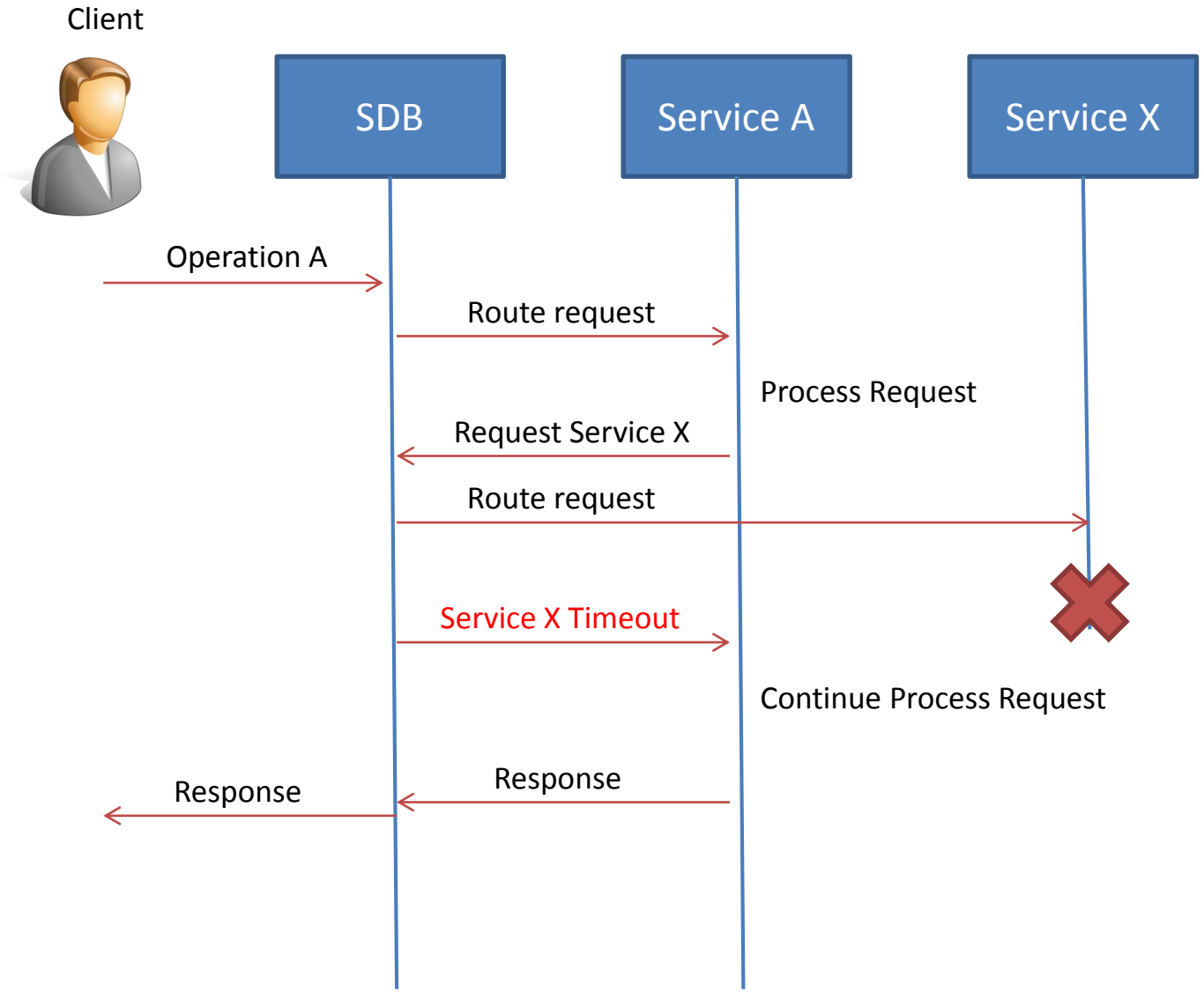
Service Management Interface

Use Cases (2/2)

- Application A publishes a ManagementReport that contains a User Interface error due to Service A inconsistent behavior. The ManagementReport is processed and a ticket is automatically opened in Problem Management and assigned to Service A owner.
- The number of occurrences of an Error Code in a set of Premium Services has raised above X% for a given time period.

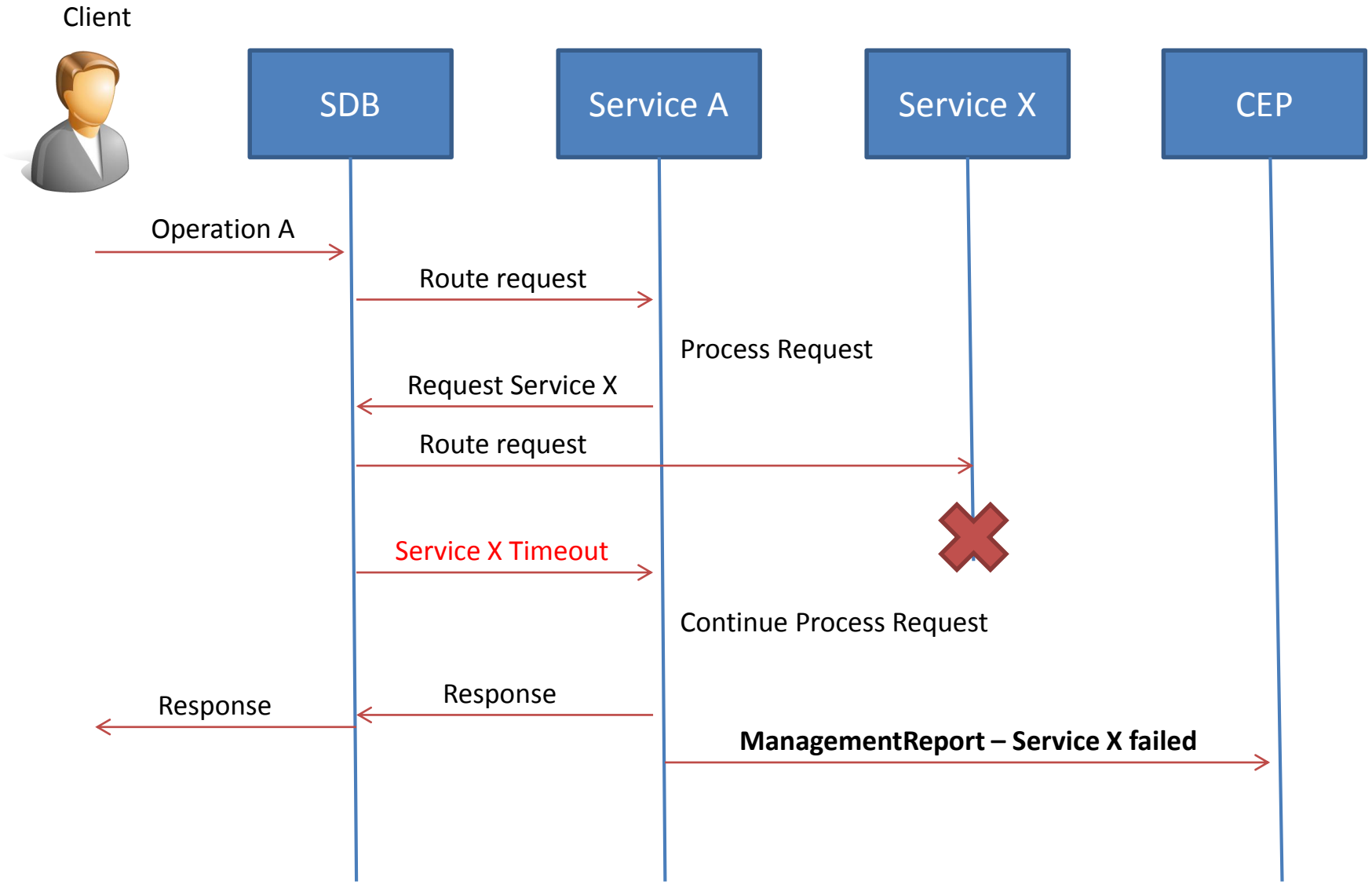
Use case

Before SMI

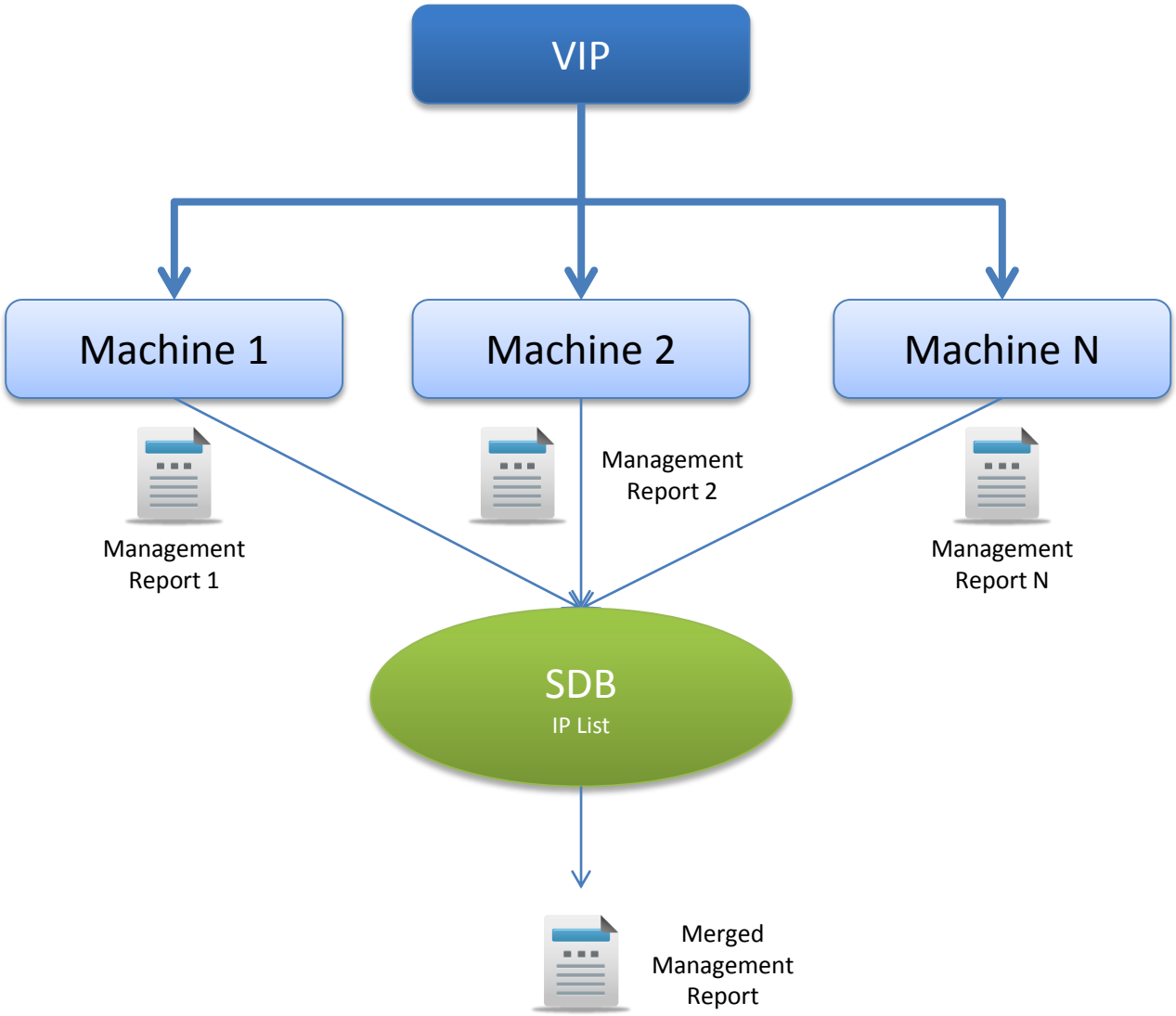


Use case

After SMI



SMI Design Example



Agenda

- Introduction
- ManagementReport
- Use Cases
- Integration Guidelines
- Takeaways

Integration Guidelines (1/2)

GetManagementReport operation

- Use the ManagementReport **type definition** available in the SDB Backoffice
  - TMForumSDFSMI
- Identify what are the resources used by your service
- Identify which failures do you want to report
- Test each resource

Integration Guidelines (2/2)

Event-Driven

- Identify the points in your code that could fail
- Colect each fail in a ManagementReport
- Send the ManagementReport
  - At each fail
  - In the end of a request

Service Management API SDK

- SMI SDK
  - .NET
  - Java

- https://github.com/sapo/sapo-services-sdk/tree/master/SMI-SDK

Agenda

- Introduction
- ManagementReport
- Use Cases
- Integration Guidelines
- Takeaways

Takeaways

- The Service Management API is simple to implement and consume.
  - Implementable on any application or service
  - Designed to be developer-friendly
- The ManagementReport entity may be used to report most management relevant data in a standardized way:
  - Metrics
  - Health/ExecutionState
  - Failures
  - Configuration Settings
- Enables a set of management patterns, principles and best practices.
  - Functional Abstraction
  - Least Privilege Access
- The SM-API implementation contributes to interesting capabilities such as: dependencies failures correlation, proactive notifications and predictive analysis, thus contributing to an higher QoS (Quality of Service)
  - The SM-API is a requirement

Service Management Interface

for Service Developers