

# OSGI REMOTE SERVICES

# OSGi Remote Services Speaker



## **Dirk Fauth**

*Software-Architect Rich Client Systeme  
Eclipse Committer*

Robert Bosch GmbH  
Franz-Oechsle-Straße 4  
73207 Plochingen

[dirk.fauth@de.bosch.com](mailto:dirk.fauth@de.bosch.com)  
[www.bosch.com](http://www.bosch.com)  
[blog.vogella.com/author/fipro/](http://blog.vogella.com/author/fipro/)  
Twitter: [fipro78](https://twitter.com/fipro78)

# MOTIVATION

# OSGi Remote Services

## Motivation

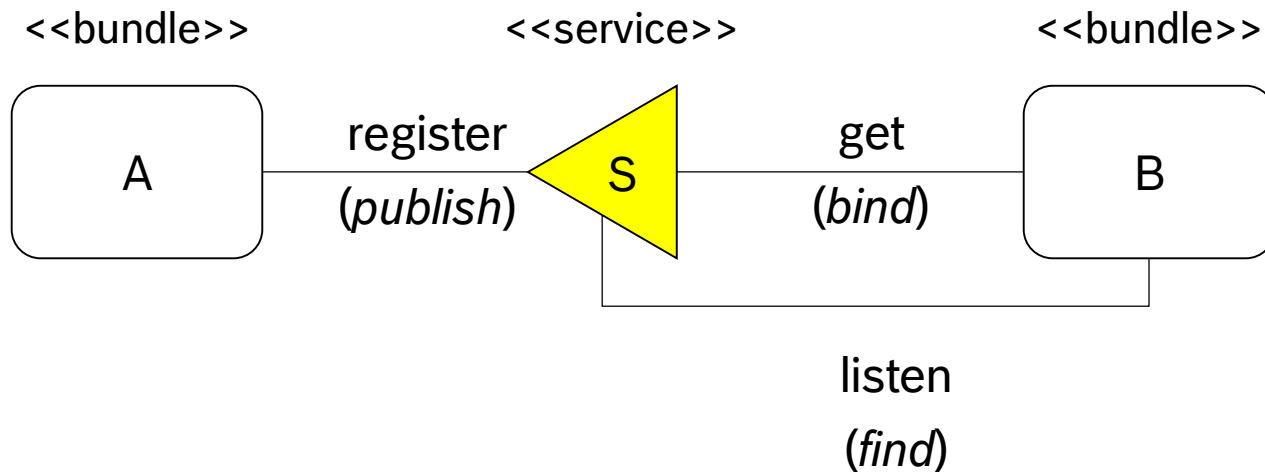
- ▶ Microservices are one of the current hype topics
- ▶ A lot of people are actually doing microservices with different approaches
- ▶ OSGi has a mature service design
  - ▶ OSGi Core R7 – Chapter 5 Service Layer
  - ▶ OSGi Compendium R7 – Chapter 112 Declarative Services
- ▶ OSGi has specifications for remote services and microservices
  - ▶ OSGi Compendium R7 – Chapter 100 Remote Services
  - ▶ OSGi Compendium R7 – Chapter 122 Remote Service Admin Service
  - ▶ (OSGi Compendium R7 – Chapter 151 JAX-RS Whiteboard)
- ▶ There are several talks about OSGi, microservices and remote services
  - ▶ But mostly about architecture, cool demos but little about the technical details

# OVERVIEW

# OSGi Remote Services

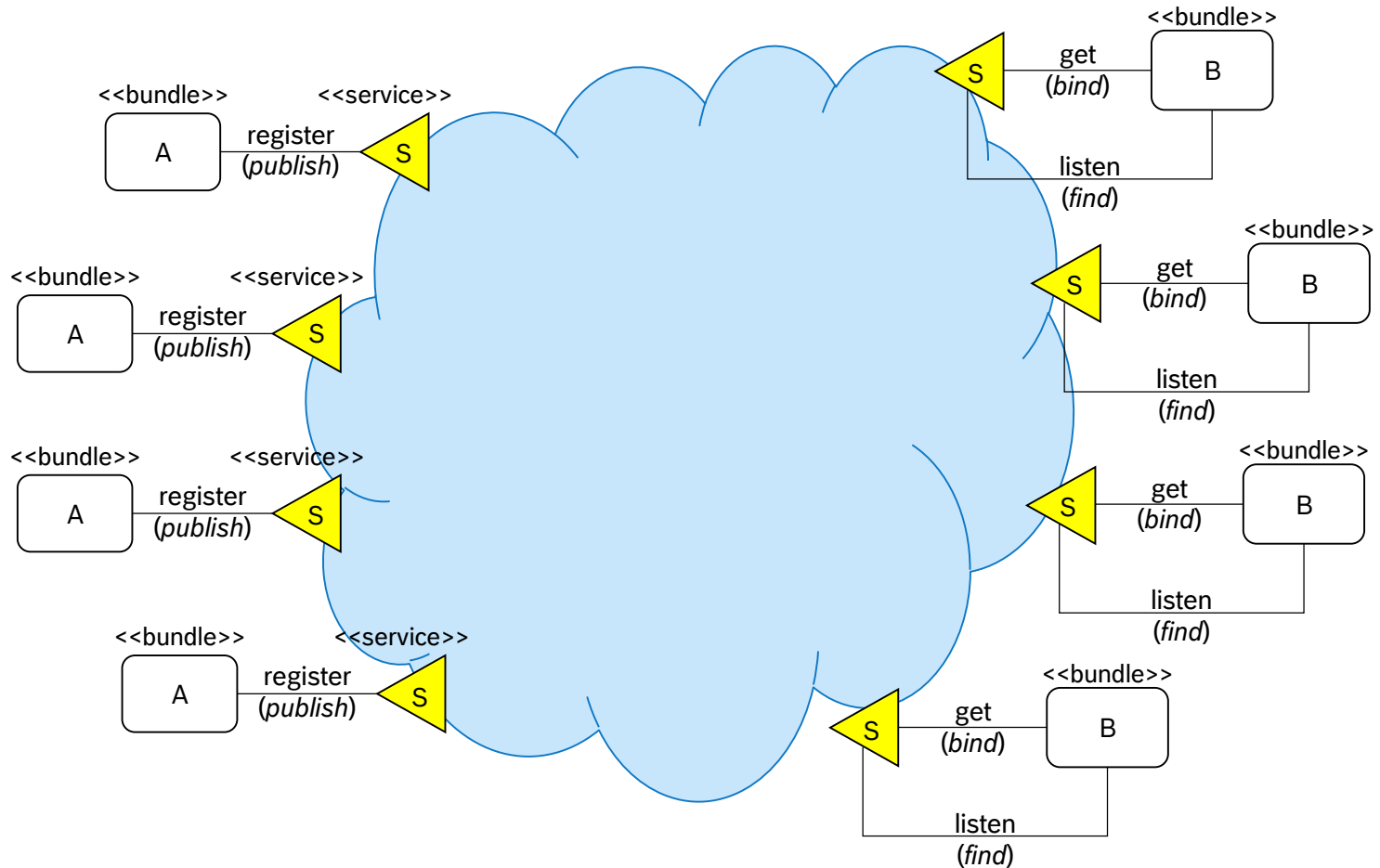
## Publish-Find-Bind

- ▶ Bundles **register (publish)** services
- ▶ Bundles **get (bind)** services
- ▶ Bundles **listen (find)** services



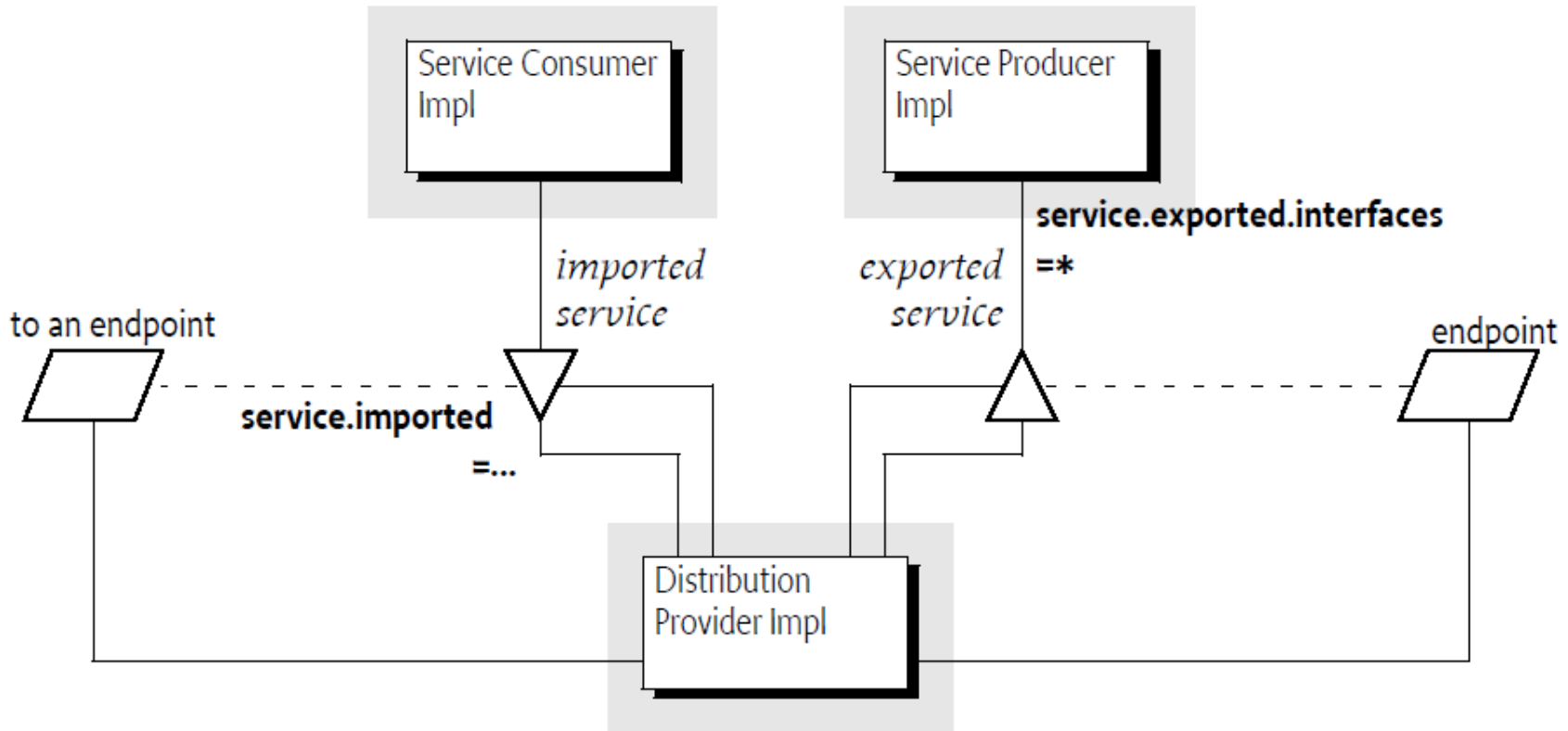
# OSGi Remote Services

## Publish-Find-Bind - Remote



# OSGi Remote Services

## Remote Services



OSGi Compendium Specification R7 – Figure 100.1 Architecture



# OSGi Remote Services

## Terminology

### ▶ **Remote Service (*aka Distributed Service*)**

- ▶ OSGi service that is available across container boundaries

### ▶ **Distribution Provider**

- ▶ Exports services by creating endpoints
- ▶ Imports services by creating proxies to access endpoints
- ▶ Manage policies around the topology
- ▶ Discover remote services

### ▶ **Endpoint**

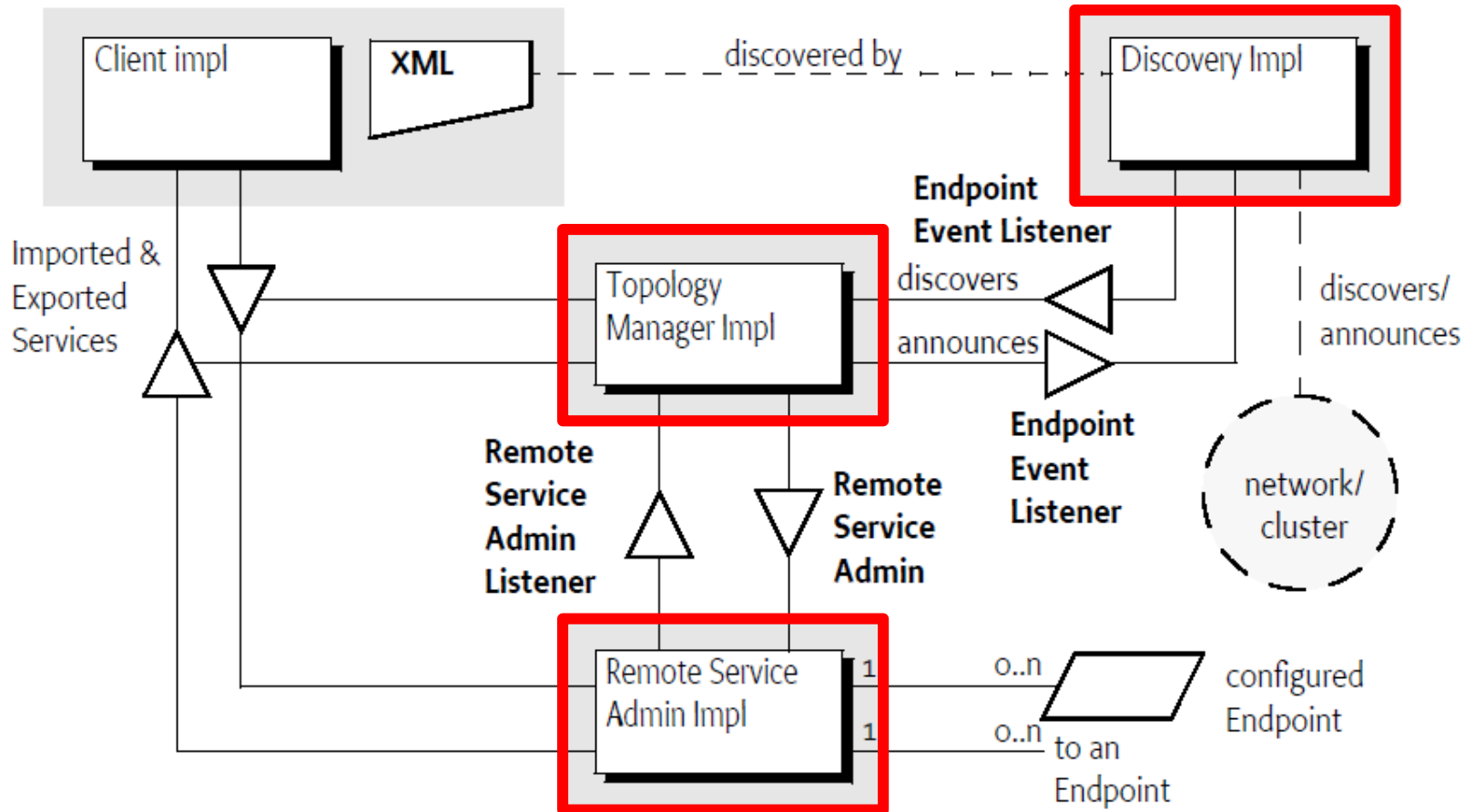
- ▶ Communication access mechanism to a remote service that requires some protocol for communications

### ▶ **Topology**

- ▶ Mapping between services and endpoints as well as their communication characteristics

# OSGi Remote Services

## Remote Service Admin



OSGi Compendium Specification R7 – Figure 122.1 Remote Service Admin Entities

# OSGi Remote Services

## Terminology

### ▶ **Remote Service Admin**

Passive Distribution Provider providing the mechanism but actually does not export or import services itself

### ▶ **Topology Manager**

Provides the policy for importing and exporting services via RSA  
Implements a topology by using the RSA

### ▶ **Discovery**

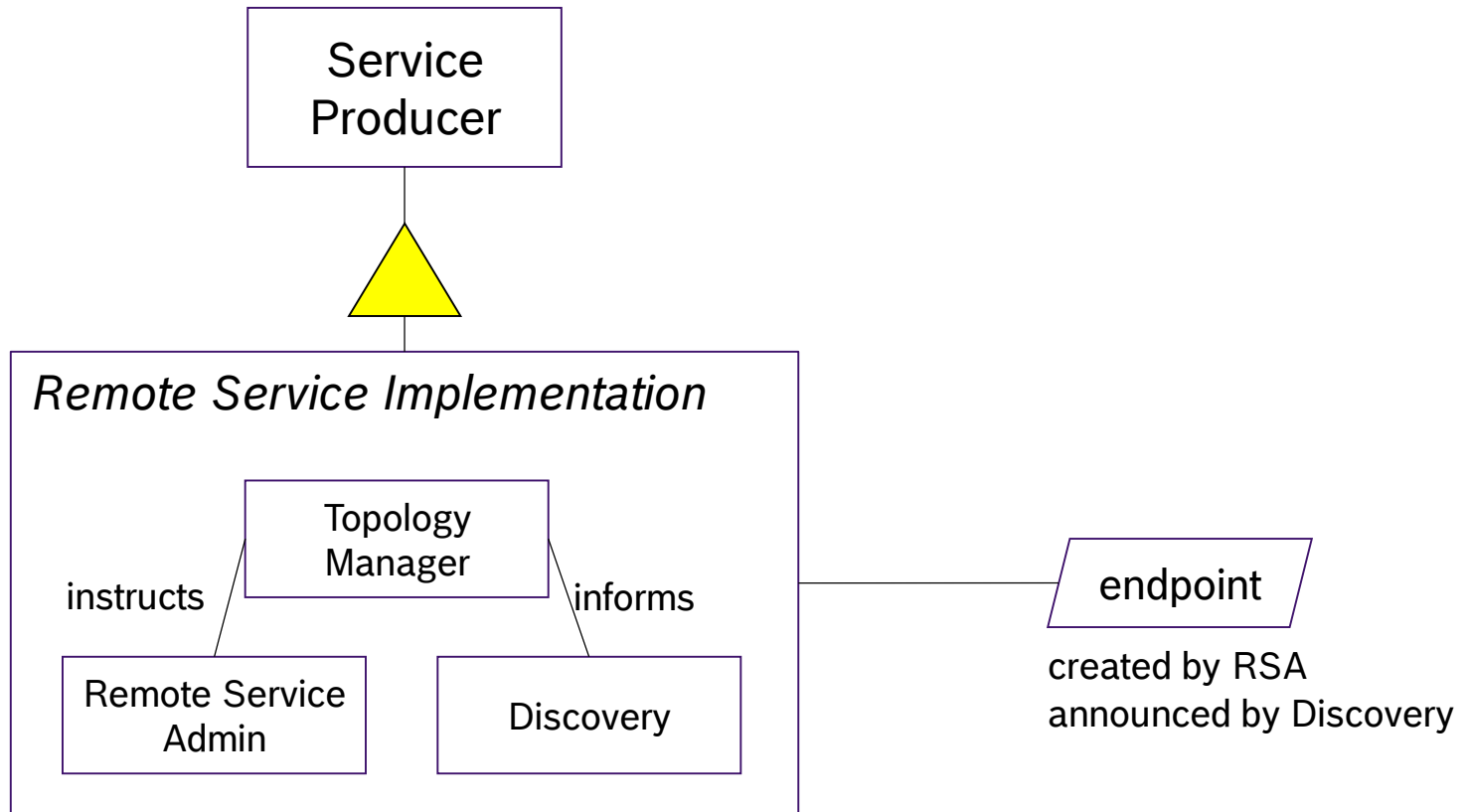
Discover / announce Endpoint Descriptions via some discovery protocol

### ▶ **Endpoint Description**

Describes an Endpoint via configuration type (name and set of properties)

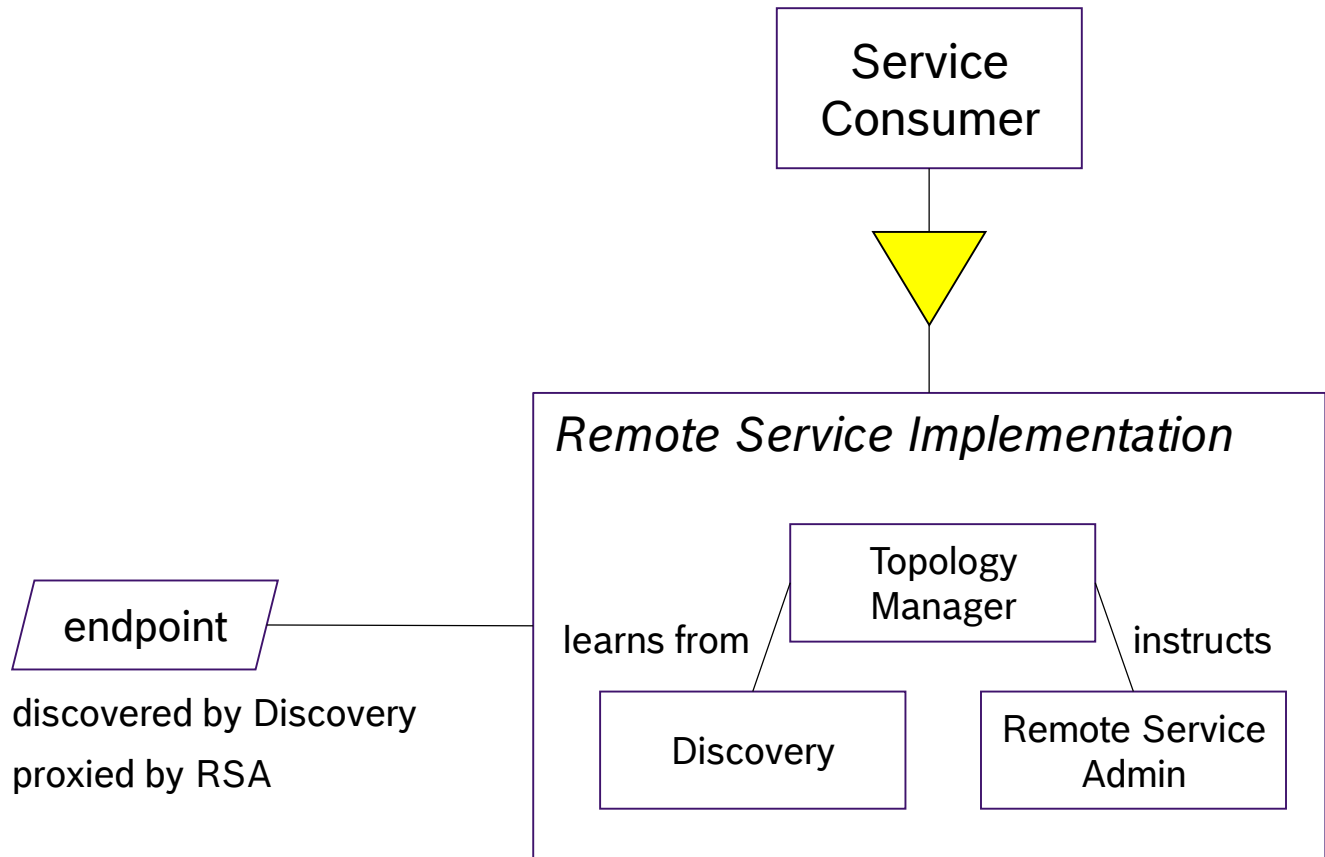
# OSGi Remote Services

## Remote Service Admin – Export Simplified



# OSGi Remote Services

## Remote Service Admin – Import Simplified



# IMPLEMENTATION

# OSGi Remote Services

## Remote Service Implementation

```
@Component(property= {
    "service.exported.interfaces=*",
    "service.exported.configs=ecf.generic.server"
})
public class UppercaseModifier implements StringModifier {

    @Override
    public String modify(String input) {
        return input.toUpperCase(Locale.getDefault());
    }

}
```

# OSGi Remote Services

## Remote Service Properties

- ▶ `service.exported.interfaces=*`  
Required property to mark a service for export. Specifies which service interfaces should be exported. Using the wildcard it says that all the interfaces that are registered should be exported.
- ▶ `service.exported.configs`  
The configuration types that should be used to export a service. Each configuration type (endpoint type) represents the configuration parameters for endpoints. For example:
  - ▶ ECF Generic Provider = `ecf.generic.server`
  - ▶ CXF DOSGi provider RS = `org.apache.cxf.rs`

[https://wiki.eclipse.org/Distribution\\_Providers](https://wiki.eclipse.org/Distribution_Providers)

<https://github.com/apache/cxf-dosgi>



# OSGi Remote Services Implementations

- ▶ Eclipse Communication Framework  
<https://www.eclipse.org/ecf/>
- ▶ Apache Aries Remote Service Admin  
<http://aries.apache.org/modules/rsa.html>
- ▶ Amdatu  
<https://amdatu.org/application/remote/>
- ▶ Paremus RSA  
<https://docs.paremus.com/display/SF113/Remote+Service+implementation>
- ▶ Apache CXF Distributed OSGi (distribution providers for Aries RSA)  
<http://cxf.apache.org/distributed-osgi.html>

# OSGi Remote Services

## Required Bundles – Equinox OSGi Application

Bundle-SymbolicName	Bundle-Name
<i>org.fipro.modifier.api</i>	<b>Application Bundles</b>
<i>org.fipro.modifier.uppercase</i>	
<i>org.apache.felix.gogo.command</i>	<b>OSGi Console</b>
<i>org.apache.felix.gogo.runtime</i>	
<i>org.apache.felix.gogo.shell</i>	
<i>org.eclipse.equinox.console</i>	
<i>org.eclipse.osgi</i>	<b>Equinox OSGi System Bundle</b>
<i>org.eclipse.osgi.services</i>	<b>Equinox OSGi Service Interfaces</b>
<i>org.eclipse.equinox.common</i>	<b>Common Eclipse Runtime</b>
<i>org.eclipse.equinox.event</i>	<b>Event Admin</b>
<i>org.eclipse.equinox.util</i>	<b>Equinox Util Bundle</b>
<i>org.apache.felix.scr</i>	<b>Apache Felix Declarative Services</b>
<i>org.eclipse.osgi.util</i>	<b>OSGi Utility Classes</b>

# OSGi Remote Services

## Required Bundles – ECF with Generic Provider

Bundle-SymbolicName	Bundle-Name
<i>org.eclipse.core.jobs</i>	Eclipse Jobs Mechanism
<i>org.eclipse.equinox.concurrent</i>	Equinox Concurrent API
<i>org.eclipse.ecf</i>	ECF Core Bundle
<i>org.eclipse.ecf.discovery</i>	ECF Discovery API Bundle
<i>org.eclipse.ecf.identity</i>	ECF Identity Bundle
<i>org.eclipse.ecf.osgi.services.distribution</i>	ECF RSA Basic Topology Manager
<i>org.eclipse.ecf.osgi.services.remoteserviceadmin</i>	ECF RSA Implementation
<i>org.eclipse.ecf.osgi.services.remoteserviceadmin.proxy</i>	ECF RSA Proxy
<i>org.eclipse.ecf.provider</i>	<b>ECF Generic Provider</b>
<i>org.eclipse.ecf.provider.jmdns</i>	<b>ECF Discovery Zeroconf/JMDNS Provider</b>
<i>org.eclipse.ecf.provider.remoteservice</i>	<b>ECF Generic Provider RemoteServices Support</b>
<i>org.eclipse.ecf.remoteservice</i>	ECF RemoteServices API
<i>org.eclipse.ecf.remoteservice.asyncproxy</i>	ECF RemoteServices AsyncProxy API
<i>org.eclipse.ecf.sharedobject</i>	ECF SharedObject API
<i>org.eclipse.osgi.services.remoteserviceadmin</i>	OSGi RSA API

# OSGi Remote Services

## ECF – Providers

### ▶ Discovery

- ▶ Zeroconf/aka Bonjour/Rendevous (JmDNS) `org.eclipse.ecf.provider.jmdns`
- ▶ jSLP aka SLP/RFC2608 `org.eclipse.ecf.provider.jslp`

### ▶ Distribution Provider

#### ▶ Generic Provider

`org.eclipse.ecf.provider`  
`org.eclipse.ecf.provider.remoteservice`

#### ▶ r-OSGi Provider

`org.eclipse.ecf.provider.r_osgi`  
`ch.ethz.iks.r_osgi.remote`

#### ▶ Apache CXF Jax-RS Implementation

`org.eclipse.ecf.provider.cxf.server`  
`org.eclipse.ecf.provider.jaxrs.server`  
`org.eclipse.ecf.provider.jaxrs`

#### ▶ ...

[https://wiki.eclipse.org/Distribution\\_Providers](https://wiki.eclipse.org/Distribution_Providers)

<https://github.com/apache/cxf-dosgi>

# OSGi Remote Services

## Troubleshooting with Equinox

- ▶ Bundles that need to auto-started
  - ▶ *org.eclipse.osgi*
  - ▶ *org.apache.felix.scr*
  - ▶ *org.eclipse.equinox.event*
  - ▶ *org.eclipse.ecf.osgi.services.distribution*
  - ▶ *org.eclipse.ecf.provider.jmdns*
  - ▶ *org.eclipse.ecf.provider.remoteservice*
- ▶ On the producer side when using a product file additional bundles needed
  - ▶ *org.eclipse.equinox.app*
  - ▶ *org.eclipse.equinox.registry*
- ▶ On the consumer side
  - ▶ Eclipse RCP and `@Service`: configure the service interface bundles for a lower start level than the default
  - ▶ Eclipse RCP with low level OSGi API: ensure the bundle is started (UI bundles are typically not started)

# OSGi Remote Services

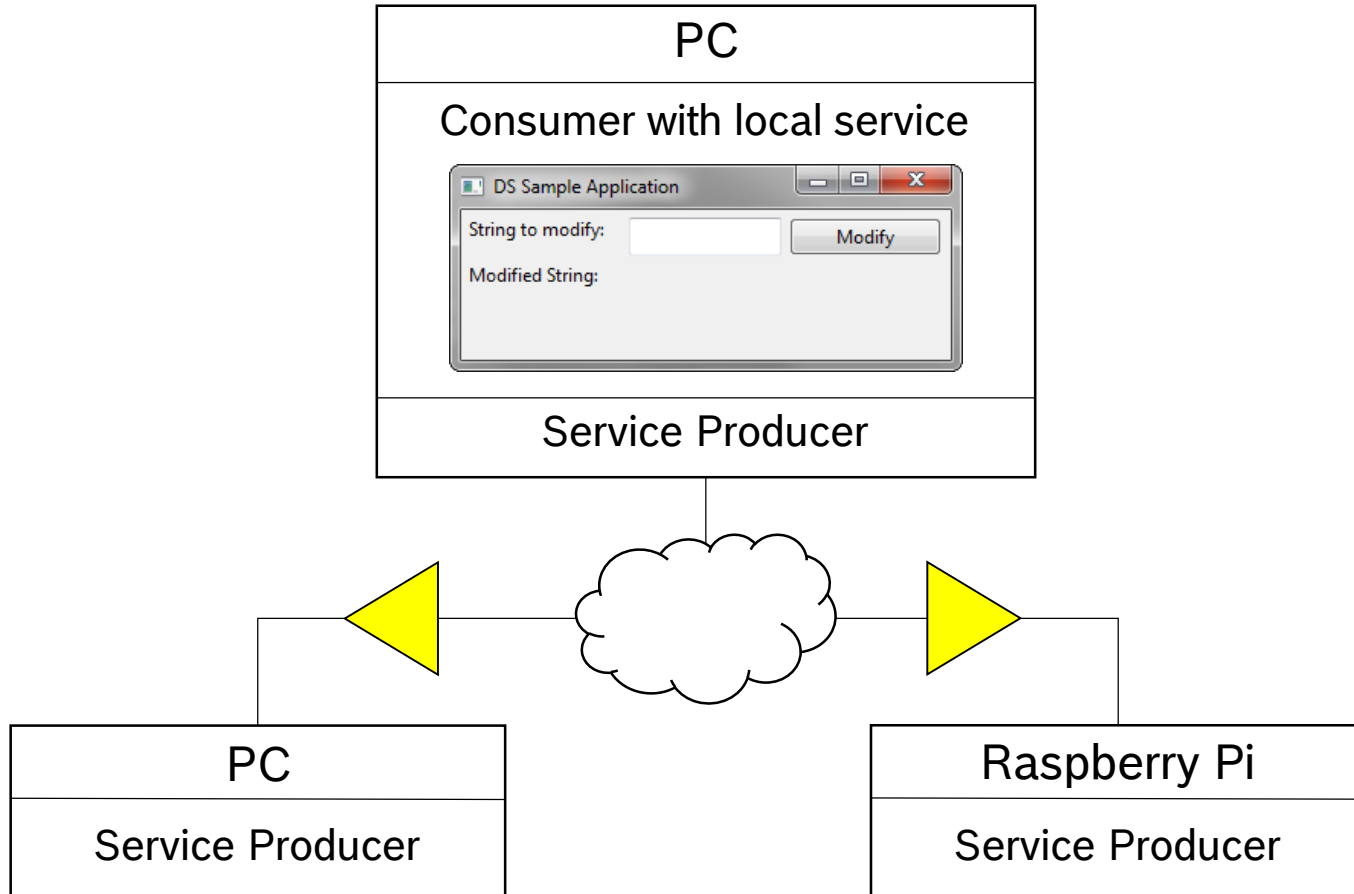
## Conclusion

- ▶ Simple at service implementation
- ▶ Complicated at runtime definition
- ▶ Even more complicated on DevOps side
  - ▶ ***Fallacies of Distributed Computing***  
[https://de.wikipedia.org/wiki/Fallacies\\_of\\_Distributed\\_Computing](https://de.wikipedia.org/wiki/Fallacies_of_Distributed_Computing)
  - ▶ ***Service interface evolution***
  - ▶ ...
- ▶ Developer specifies topology via service property, not the administrator

# DEMO

# OSGi Remote Services

## Demo Setup





# REFERENCES

# OSGi Remote Services

## References

- ▶ OSGi Core Release 7 Specification  
<https://osgi.org/specification/osgi.core/7.0.0/>
- ▶ OSGi Compendium Release 7 Specification  
<https://osgi.org/specification/osgi.cmpn/7.0.0/>
- ▶ Eclipse Communication Framework Wiki  
[https://wiki.eclipse.org/Eclipse\\_Communication\\_Framework\\_Project#OSGi\\_Remote\\_Services](https://wiki.eclipse.org/Eclipse_Communication_Framework_Project#OSGi_Remote_Services)

# OSGi Remote Services

## References

- ▶ Neil Bartlett - Scaling and Orchestrating Microservices with OSGi  
<https://de.slideshare.net/mfrancis/scaling-and-orchestrating-microservices-with-osgi-n-bartlett>
- ▶ Achim Nierbeck – Microservices and OSGi – running with Apache Karaf  
<https://de.slideshare.net/AchimNierbeck/microservices-osgirunningwithapachekaraf>
- ▶ Christian Schneider – Lean microservices on OSGi  
<https://de.slideshare.net/ChristianSchneider3/lean-microservices-on-osgi>
- ▶ Graham Charters - Microservices & OSGi - Better Together?  
<https://de.slideshare.net/mfrancis/microservices-osgi-better-together-graham-charters>

# OSGi Remote Services

## References

- ▶ Wim Jongman - How to cook an egg with the Eclipse Communication Framework  
<http://www.eclipsecon.org/europe2014/sites/default/files/slides/How%20to%20Cook%20an%20Egg%20with%20the%20Eclipse%20Communication%20Framework.pdf>
- ▶ Christoph Keimel – Powering a Live Escape Game with ECF & e(fx)clipse  
<https://de.slideshare.net/keimel/powering-a-live-escape-game-with-ecf-and-efxclipse>