Distributed OSGi/RFC 119 – The ECF way

Markus Alexander Kuppe

http://www.eclipse.org/ecf

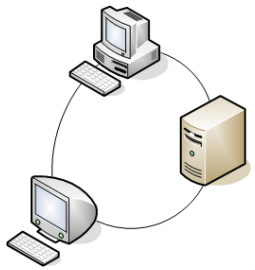
Roadmap

1. ECF in two slides
2. Architectural overview *RFC 119 – Distributed OSGi*
3. Demo (Screencast) of 119 in action

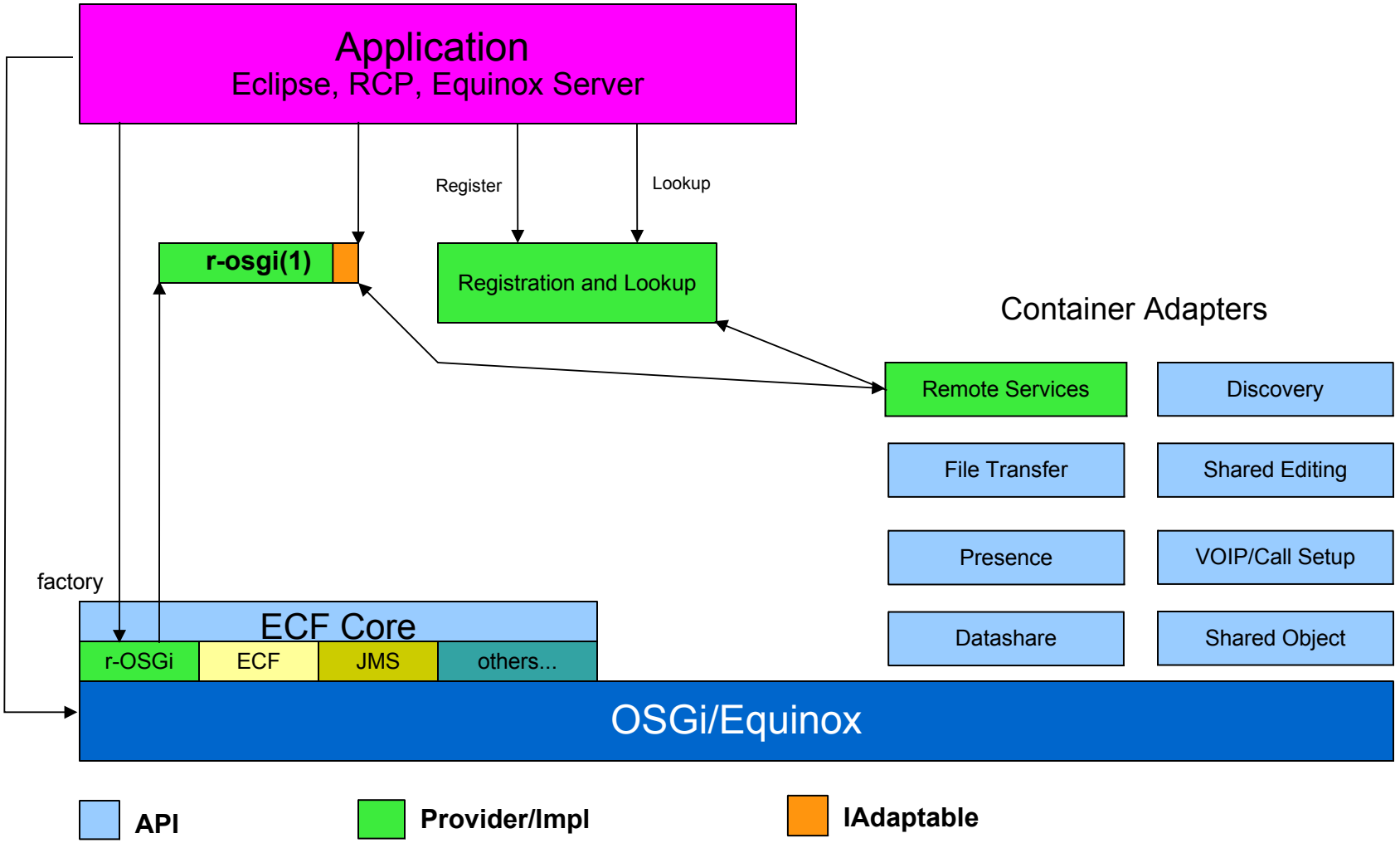
ECF: Eclipse Communication Framework

- Communication platform of Eclipse
  - Eclipse Runtime project

- Mission
  - Support team and community collaboration
  - In combination with the Eclipse IDE
    - Shared editing, file transfer, messaging
  - **As an interprocess communication platform for OSGi apps**
    - **e.g., service discovery, remote services**



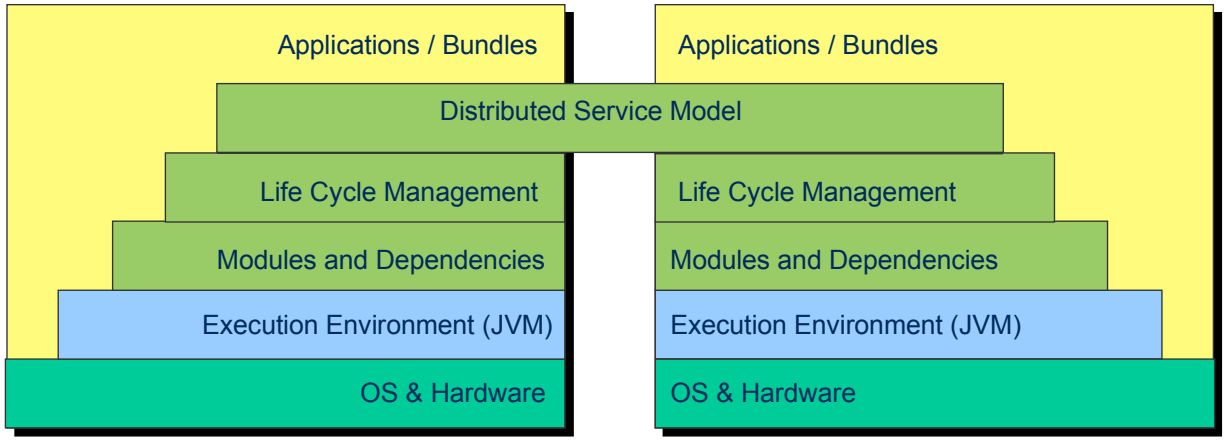
ECF: Provider Architecture



Distributed OSGi: Goals

Mission

- **Extend OSGi standard by distributed computing capabilities**

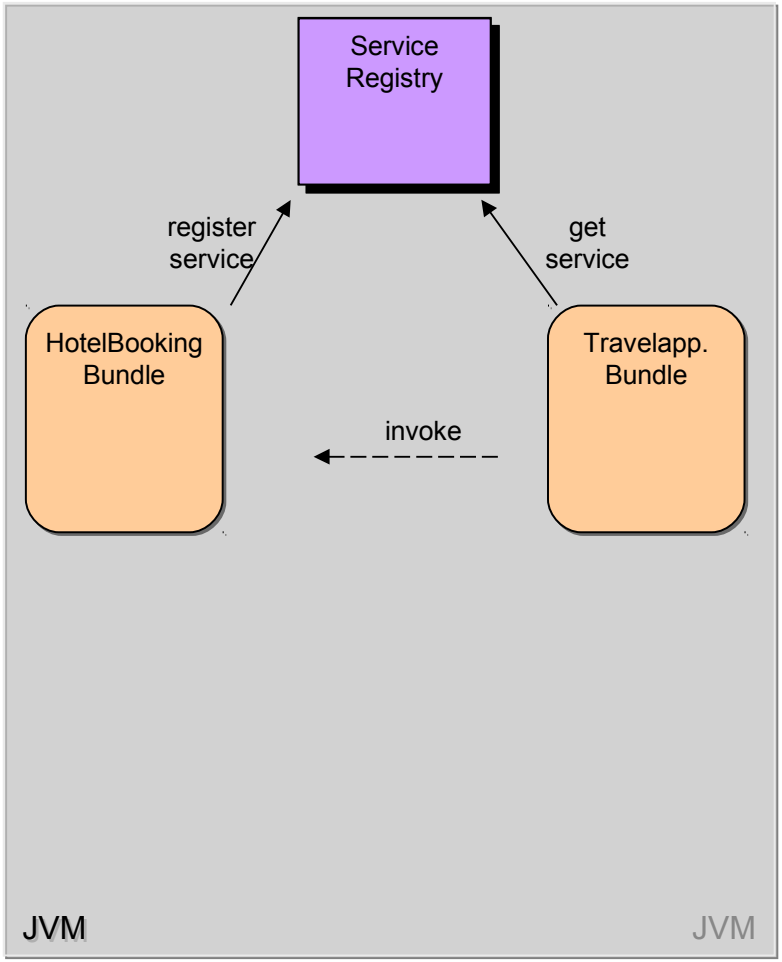


Key Requirements for Distributed OSGi

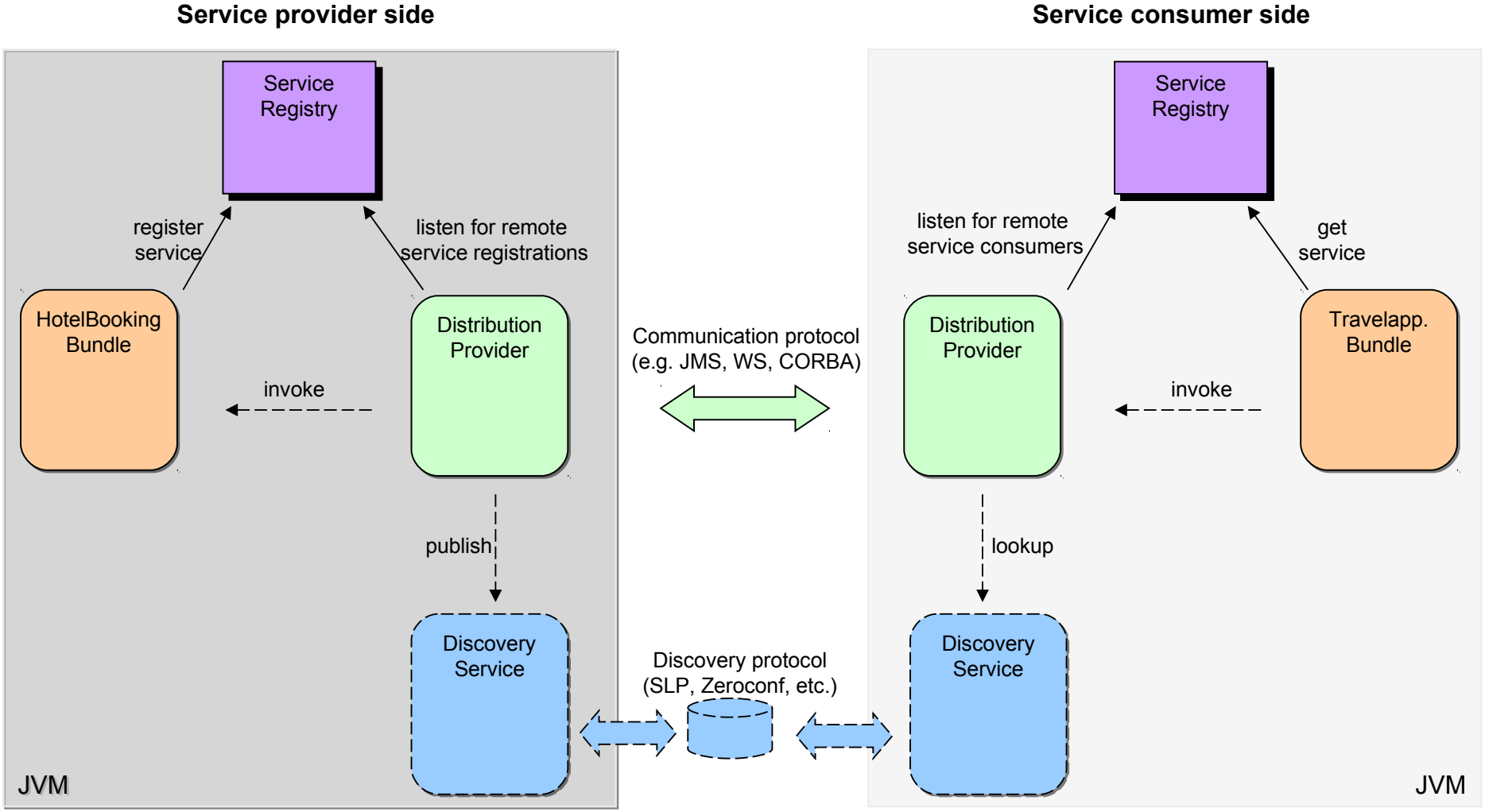
- Keep to the existing OSGi programming model where possible
- Allow abstraction from communication protocols, data formats etc.
- Allow interoperability with Non-OSGi services & clients
- Allow discovery of remote services and their usage

Distributed OSGi: Architecture

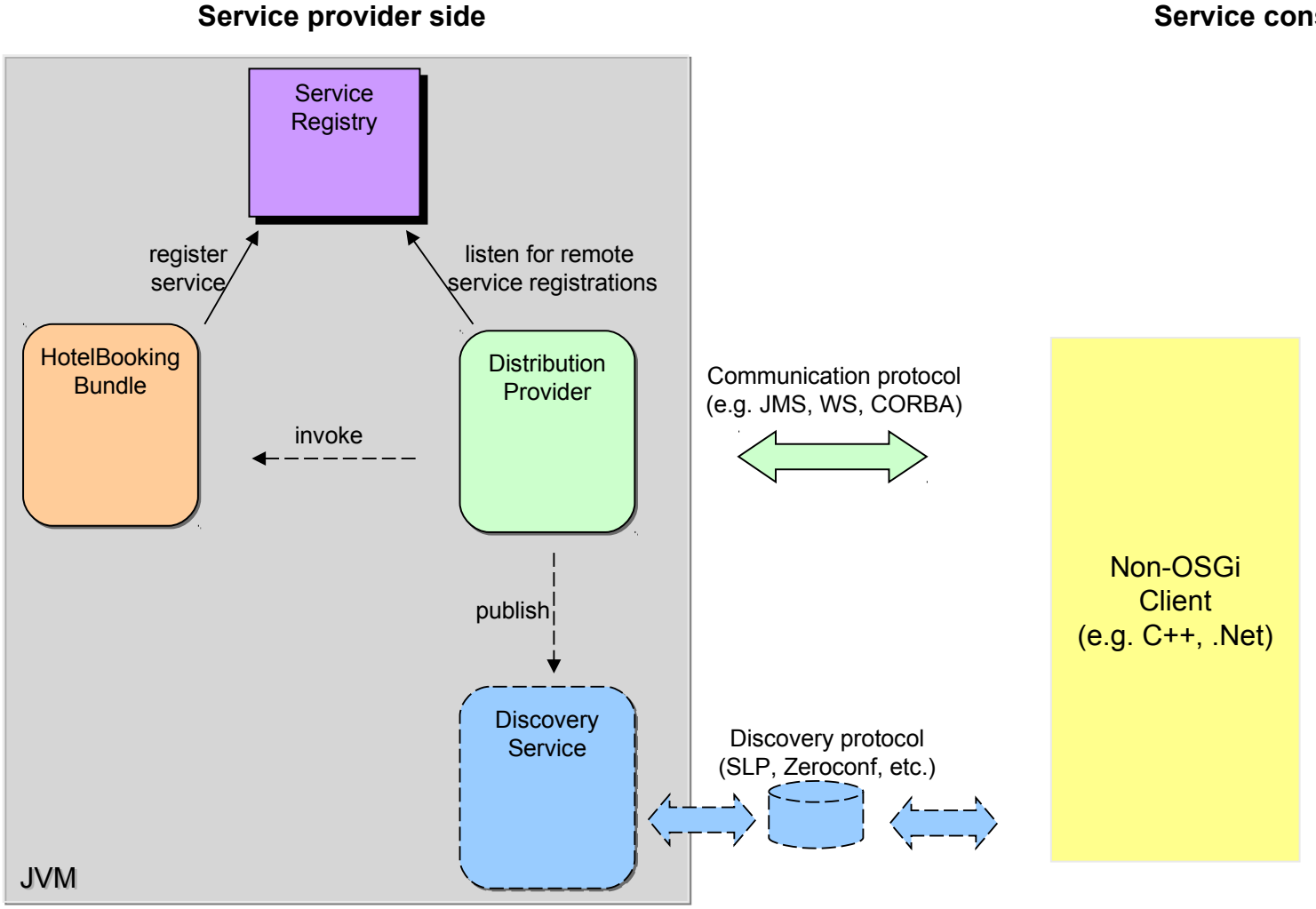
Service provider side **OSGi service model today:** **Service consumer side**



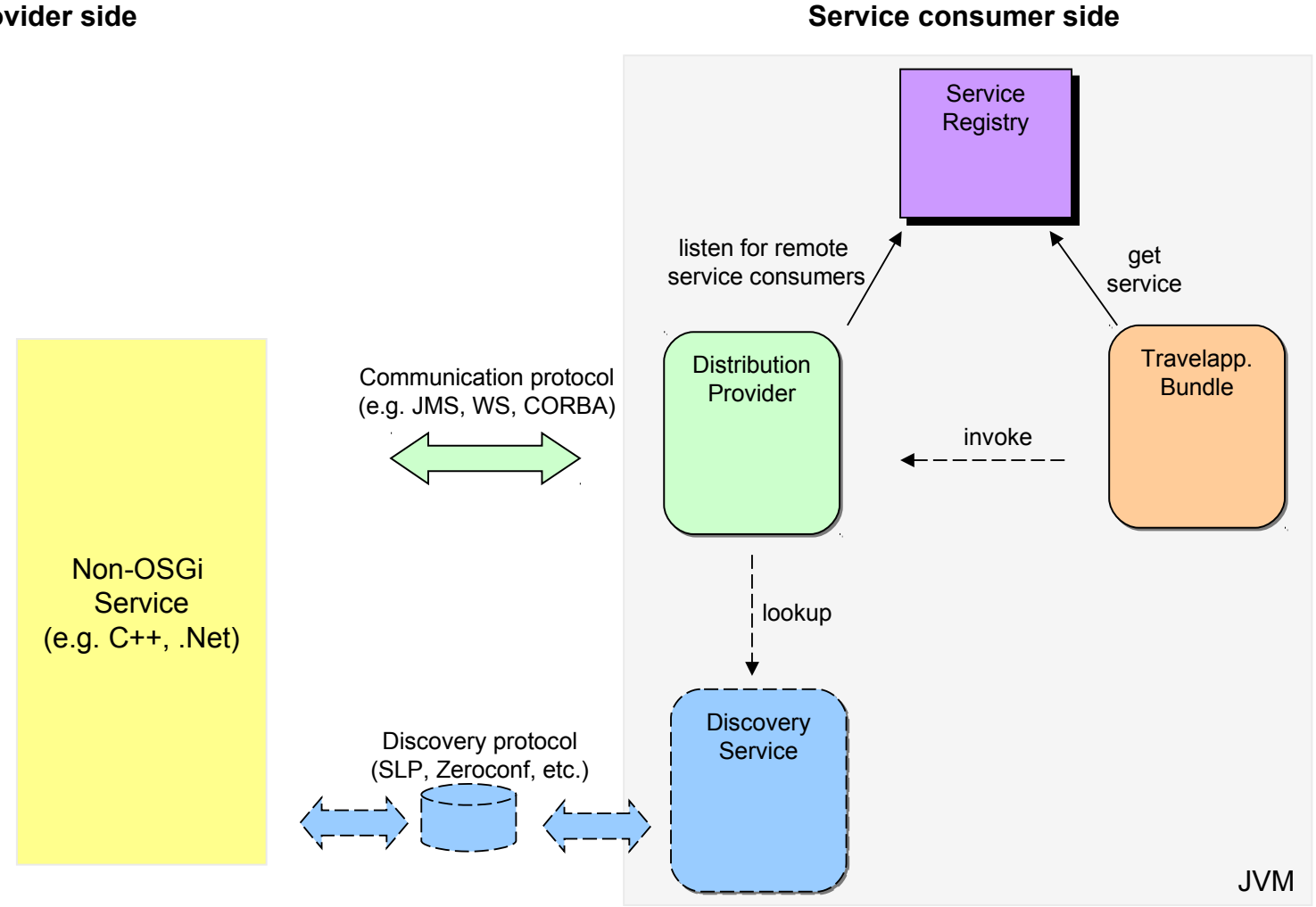
Distributed OSGi: Architecture



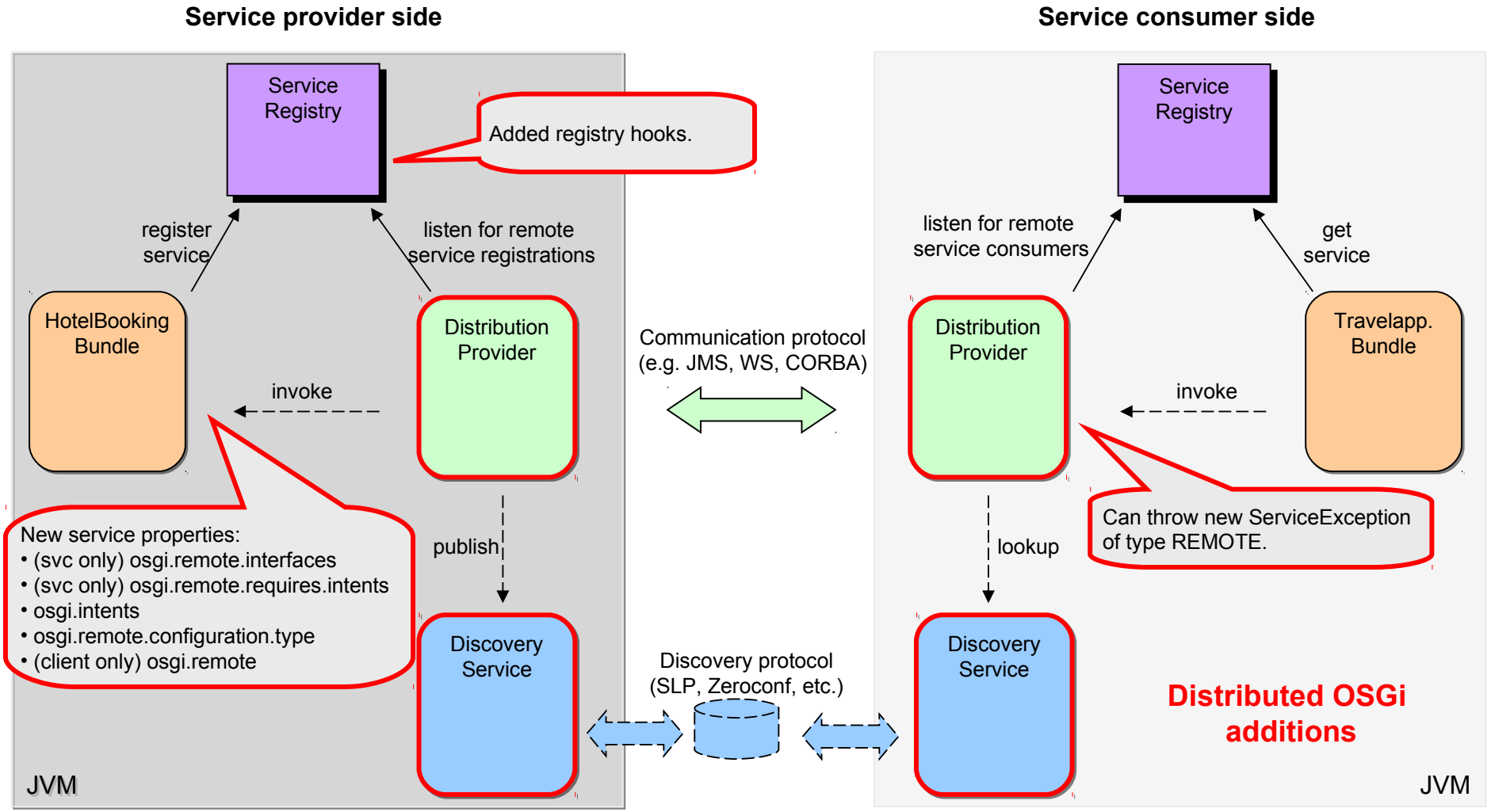
Distributed OSGi: Interoperability



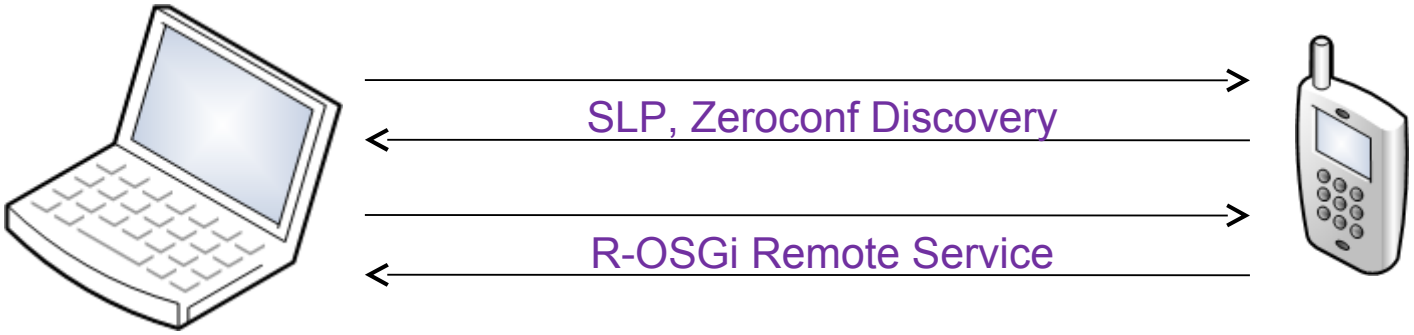
Distributed OSGi: Interoperability



Distributed OSGi: Architecture



Demo



Client

running RCP App with ECF Discovery (using SLP and mDNS providers) and distributed OSGi/RFC119 (using R-OSGi Provider)

iPhone

running Equinox and ECF, featuring an RemoteEnvInfo service remoted via RFC 119

Recap

- RFC119 provide transparent remote service registration, lookup, and clean-up
  - Makes Registration, Lookup, Clean-up work for programmers
- Framework handles dynamic services **already** so network services are accomodated
  - Caveat Emptor: Service programmers **must** make their bundles/services dynamic-aware
- But there is also **Usage** Transparency
  - public **MyResult** foo(**MyParameter**);
  - Call by value/Call by reference
  - Clients will expect it to work when they call it
  - It's going to fail with RuntimeException...or worse, **block**
  - **...**

Define your service interface 'carefully'

Eclipse ECF Project

- http://www.eclipse.org/ecf
- http://wiki.eclipse.org/ECF

- ECF 3.0 will ship with Eclipse Galileo
  - Call for suggestions: https://bugs.eclipse.org/270652

Questions?

Legal Notices

- EclipseSource logo and trademarks are registered
- IBM and the IBM logo are trademarks or registered trademarks of IBM Corporation, in the United States, other countries or both.
- Java and all Java-based marks, among others, are trademarks or registered trademarks of Sun Microsystems in the United States, other countries or both.
- OSGi is a trademark or registered trademark of the OSGi Alliance in the United States and other countries.
- Eclipse and the Eclipse logo are trademarks of Eclipse Foundation, Inc.
- Other company, product and service names may be trademarks or service marks of others.