



Dubbo框架扩展

2011-09 梁飞

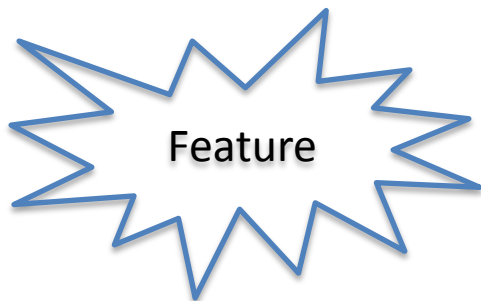
Alibaba-B2B-Platform-Framework

2011-09 梁飞



What can I do?

- Enhancement
 - monitor
 - security
 - control
 - convert
 - ...
- Optimization
 - QoS
 - asm
 - protobuf
 - grizzly
 - ...
- Integration
 - napoli
 - aranda
 - isearch
 - memcached
 - ...





Outline

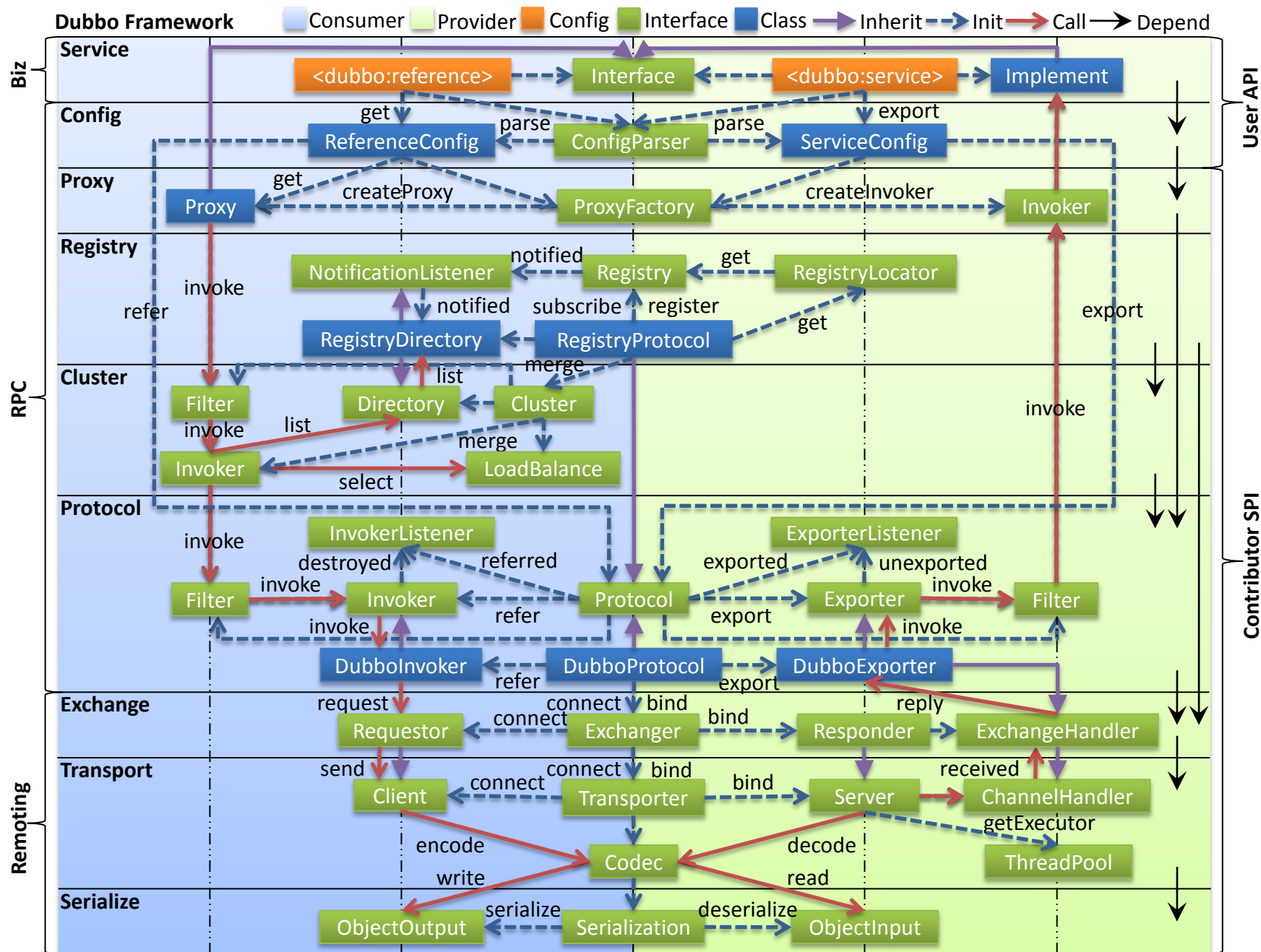
Framework

- Layers & Modules

SPI

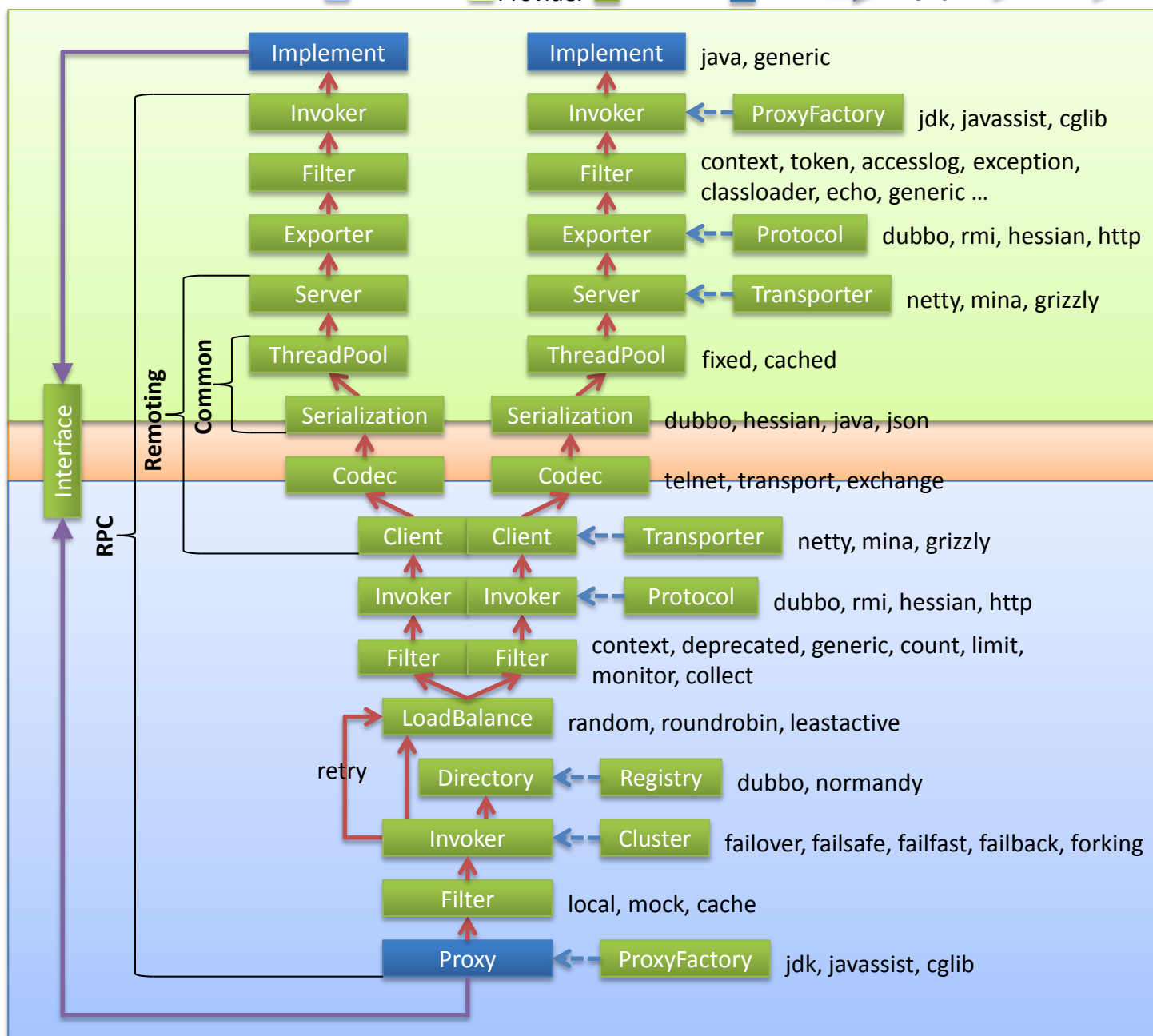
API





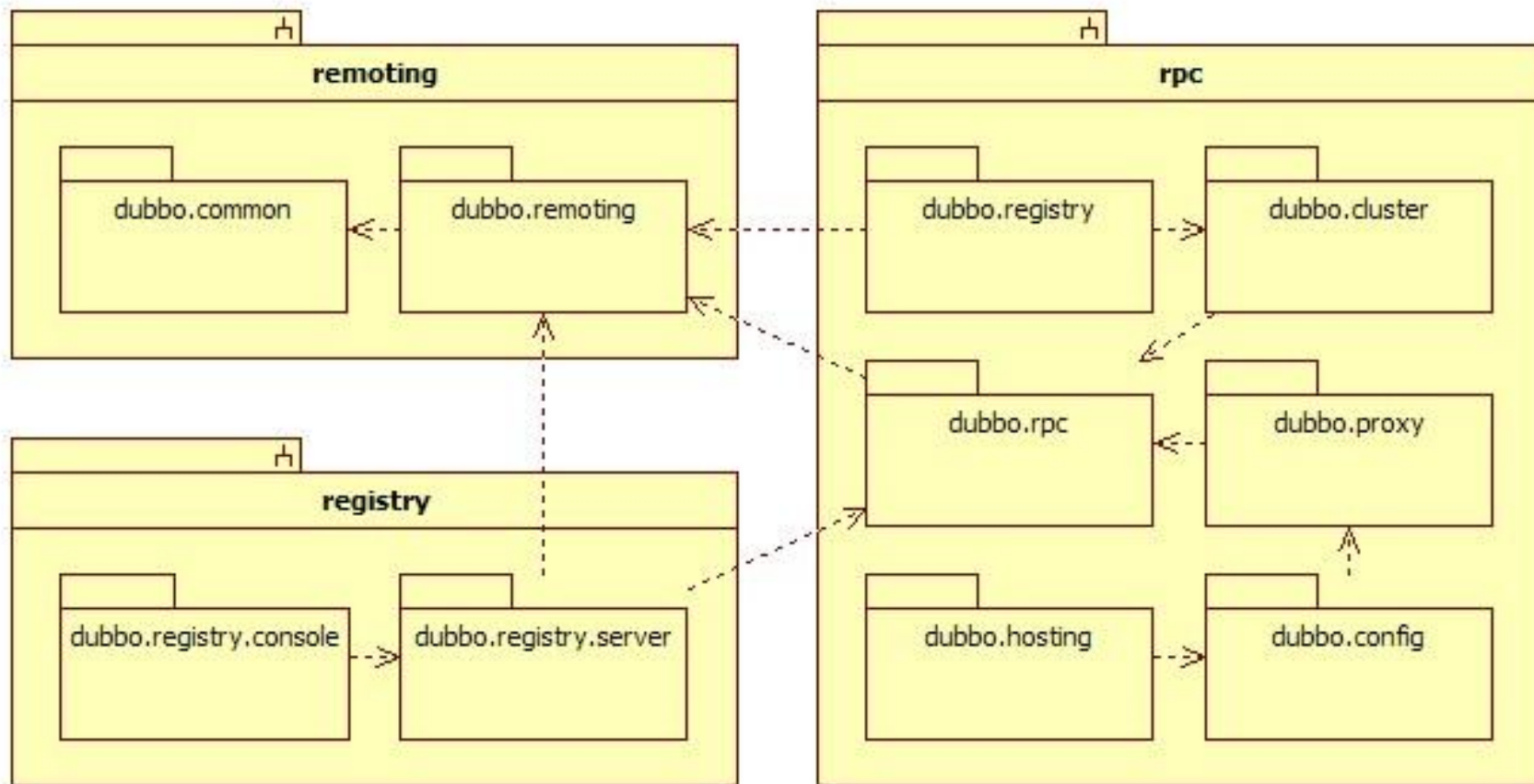
Dubbo Extension

Consumer Provider Interface Class Inherit Init Call





Modules





Outline

Framework



SPI

- Microkernel & SPI
- Protocol & ProxyFactory & Filter
- Cluster & Directory & Router & LoadBalance
- Transporter & Serialization & ThreadPool
- TelnetHandler & StatusChecker



API



Microkernel (Lifecycle Manager)

- OSGi
 - Equinox
 - Eclipse, HSF
 - META-INF/MANIFEST.MF
- IoC
 - Spring
 - META-INF/spring/beans.xml
- SPI
 - java.util.ServiceProvider
 - JDBC, MessageDigest, ScriptEngine
 - META-INF/services/com.xxx.Xxx





SPI Contract

```
src
|-main
|-java
|-com
|-xxx
|-XxxProtocol.java
|-resources
|-META-INF
|-services
|-com.alibaba.dubbo.rpc.Protocol
```

```
package com.xxx;
import com.alibaba.dubbo.common.Extension;
import com.alibaba.dubbo.rpc.Protocol;
@Extension("xxx")
public class XxxProtocol implements Protocol {
    // ...
}
```

com.xxx.XxxProtocol

- **Contract:**
 - Singleton-instance
 - Thread-safe
- **Reference:**
 - <http://download.oracle.com/javase/1.4.2/docs/guide/jar/jar.html#Service%20Provider>
 - <http://download.oracle.com/javase/6/docs/api/java/util/ServiceLoader.html>



IoC & AOP

- IoC by setter

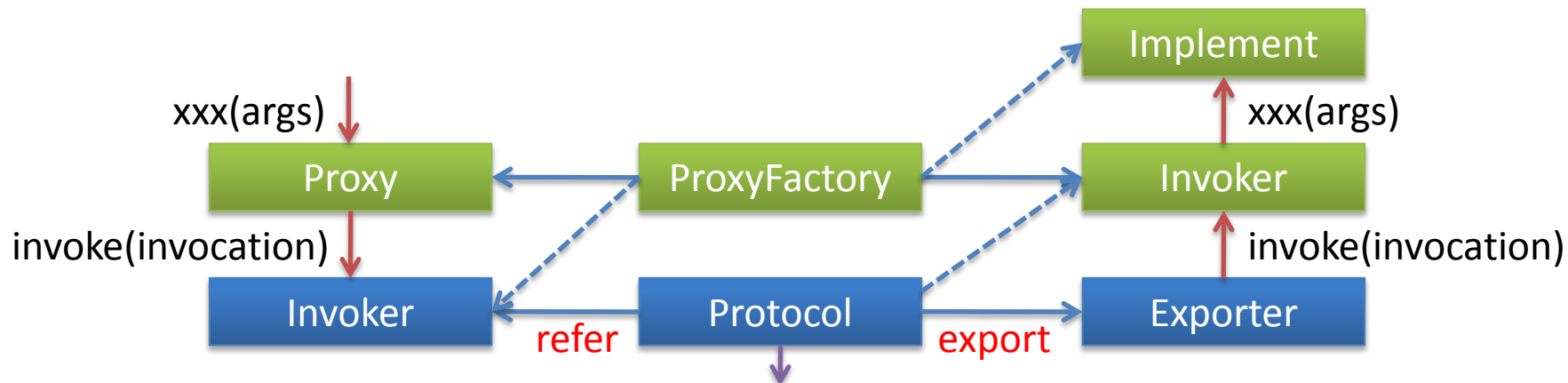
```
public XxxCluster implements Cluster {  
    private LoadBalance loadbalance;  
    public void setLoadBalance(LoadBalance loadbalance) {  
        this.loadbalance = loadbalance;  
    }  
}
```

- AOP by constructor

```
public XxxCluster implements Cluster {  
    private Cluster cluster;  
    public XxxCluster(Cluster cluster) {  
        this.cluster = cluster;  
    }  
}
```



Protocol



```
@Extention("xxx")
public class XxxProtocol implements Protocol {
    public <T> Exporter<T> export(Invoker<T> invoker) {
        return new XxxExporter(invoker);
    }
    public <T> Invoker<T> refer(Class<T> type, URL url) {
        return new XxxInvoker(type, url);
    }
}
```

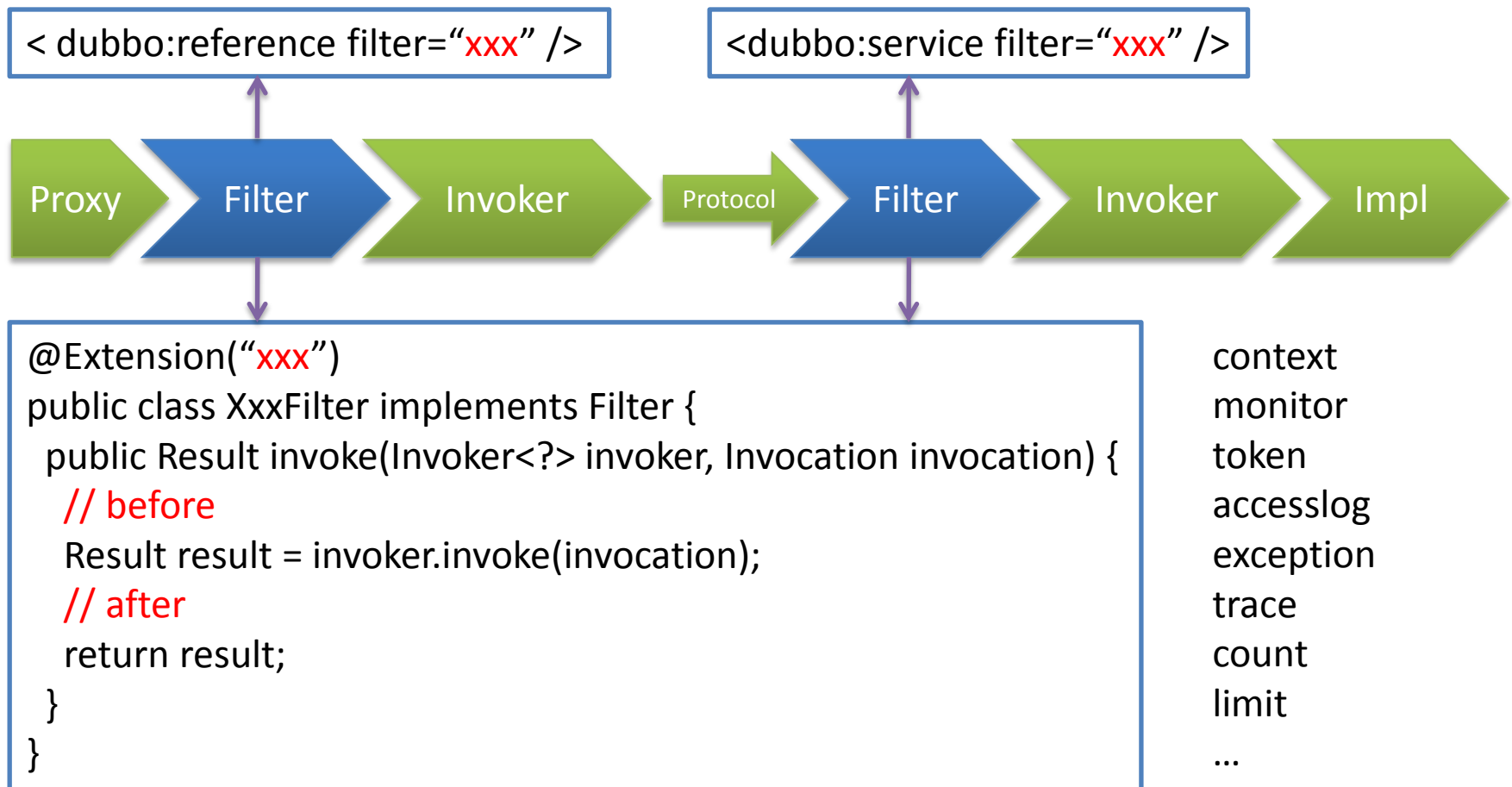
dubbo
rmi
hession
http

napoli
aranda
isearch
memcached

```
<dubbo:protocol name="xxx" port="1234" />
```

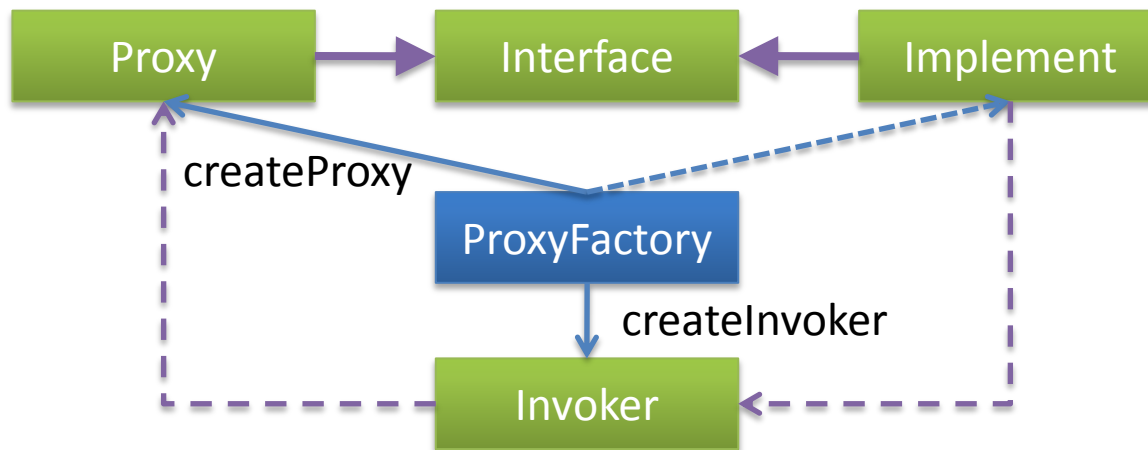


Filter





ProxyFactory



```
@Extention("xxx")
public class XxxProxyFactory implements ProxyFactory {
    public <T> T createProxy(Invoker<T> invoker, Class<?>... types) {
    }
    public <T> Invoker<T> createInvoker(T instance, Class<T> type, URL url) {
    }
}
```

```
<dubbo:reference proxy="javassist" />
```



JDK vs. Javassist

- Proxy

```
Proxy.newProxyInstance(interfaceClass.getClassLoader(), new Class [] {interfaceClass}, new RemoteInvocationHandler());
```



```
public class Proxy$1 implements XxxService {  
    // .....  
}
```

- Invoker

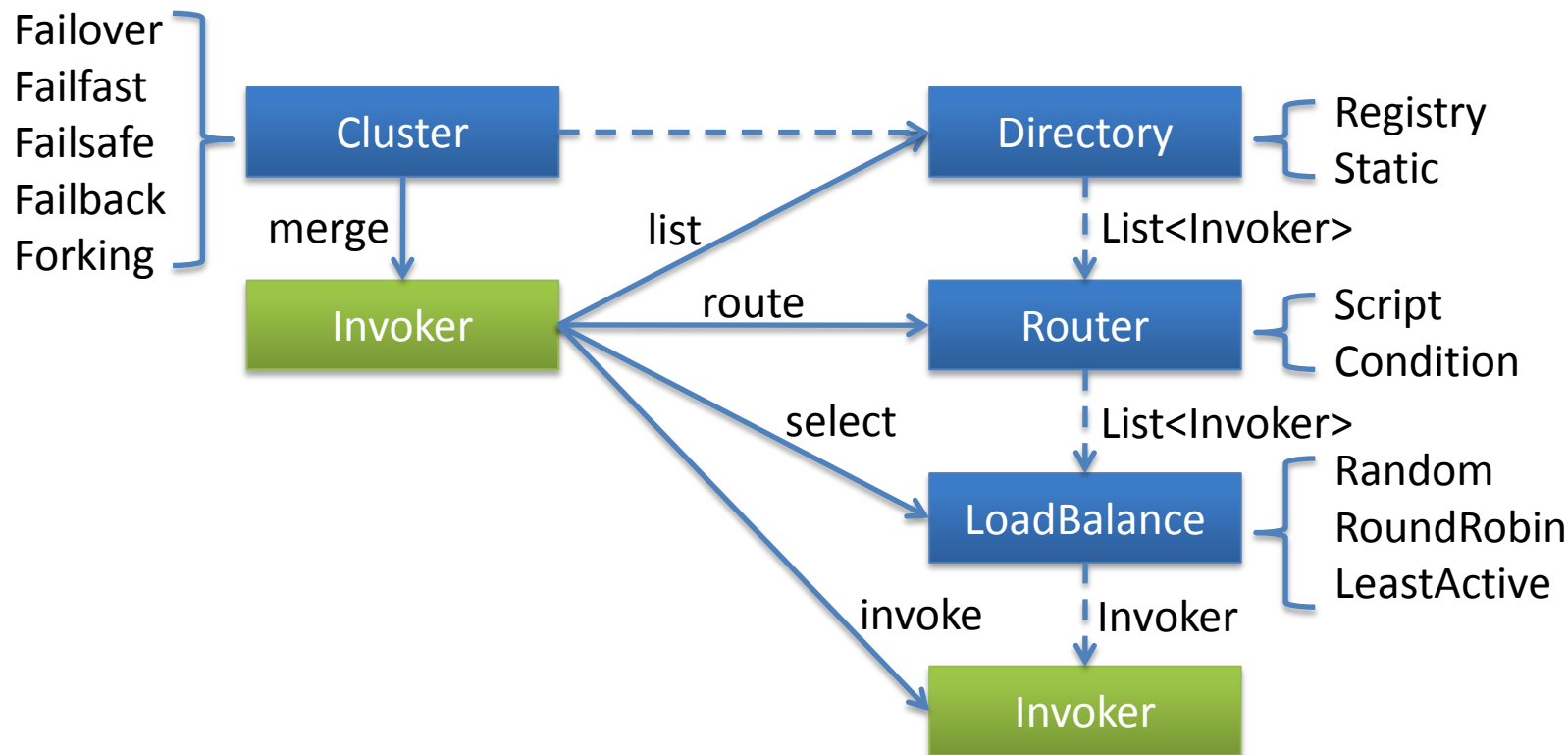
```
impl.getClass().getMethod(methodName, parameterTypes).invoke(impl, args);
```



```
public class Invoker$1 implements Invoker {  
    public Object invoke(String methodName, Class[] parameterTypes, Object[] args) throws Exception {  
        if ("xxx".equals(methodName)) {  
            impl.xxx(args[0], args[1]);  
        } else if ("yyy".equals(methodName)) {  
            impl.yyy(args[0]);  
        } // .....  
    }  
}
```



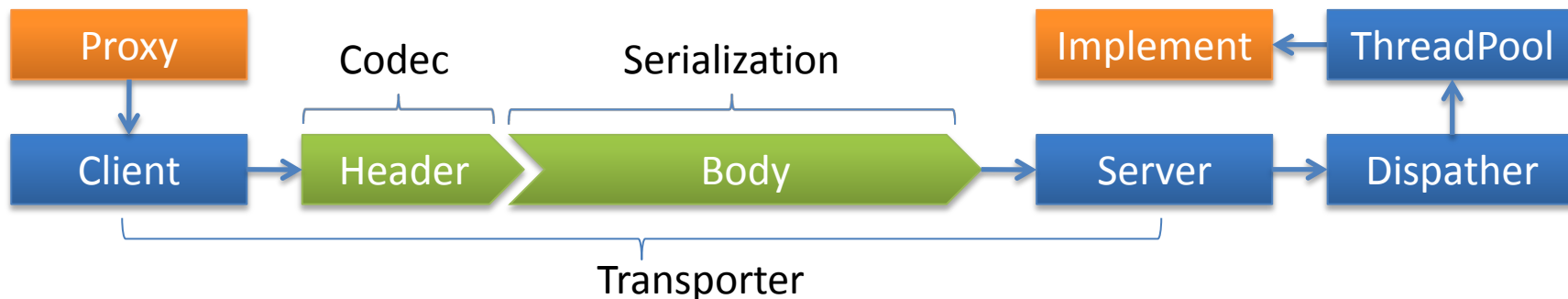
Cluster



```
<dubbo:service cluster="failover" router="script" loadbalance="random" />
<dubbo:reference cluster="failover" router="script" loadbalance="random" />
<dubbo:registry address="10.20.153.186:9090" />
<dubbo:reference url="registry:// 10.20.153.186:9090" />
<dubbo:reference url="dubbo://10.20.153.10:20880;dubbo://10.20.153.11:20880" />
```



Remoting



- Transporter
 - `<dubbo:protocol name="dubbo" client="netty" server="netty" />`
 - netty, mina, grizzly
- Serialization
 - `<dubbo:protocol name="dubbo" serialization="hessian2" />`
 - dubbo, hessian2, java, json
- ThreadPool
 - `<dubbo:protocol name="dubbo" threadpool="fixed" threads="100" />`
 - fixed, cached



Telnet

```
C:\> telnet localhost 20880  
dubbo> help
```

- ls -l
- ps -l
- status -l
- trace XxxService 10
- count XxxService
- invoke XxxService.xxxMethod(args)
- log 100
- help



TelnetHandler & StatusChecker

```
@Extension("xxx")
public class XxxTelnetHandler implements TelnetHandler {
    public String telnet(String command) throws RemotingException {
        return "echo: " + command;
    }
}
```

```
dubbo> xxx hello
echo: hello
```

```
@Extension("xxx")
public class XxxStatusChecker implements StatusChecker {
    public Status check() {
        return new Status(Status.Level.OK, "Fine!");
    }
}
```

```
dubbo> status -l
| Xxx | OK | Fine! |
```

```
<dubbo:protocol name="dubbo" telnet="xxx" status="xxx" />
```



Outline

Framework

SPI

API

- Config & URL
- Context & Generic & Local

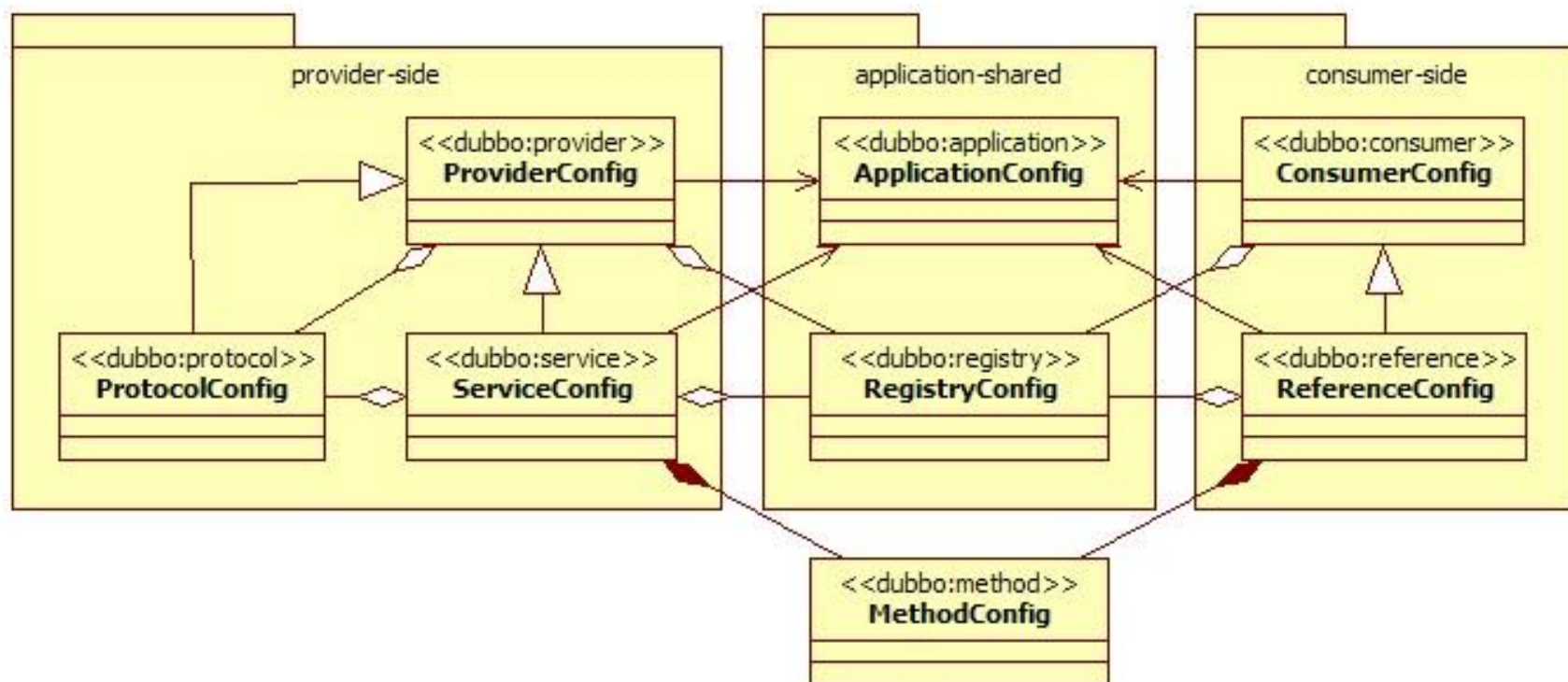


Configs

- Application Shared:
 - **<dubbo:application name="xxx" />**
 - **<dubbo:registry address="10.20.153.186" />**
- Provider Side:
 - **<dubbo:protocol name="dubbo" port="20880" />**
 - **<dubbo:service interface="com.xxx.XxxService" />**
 - **<dubbo:provider protocol="dubbo" />**
- Consumer Side:
 - **<dubbo:reference interface="com.xxx.XxxService" />**
 - **<dubbo:consumer timeout="5000" />**



Config Relation



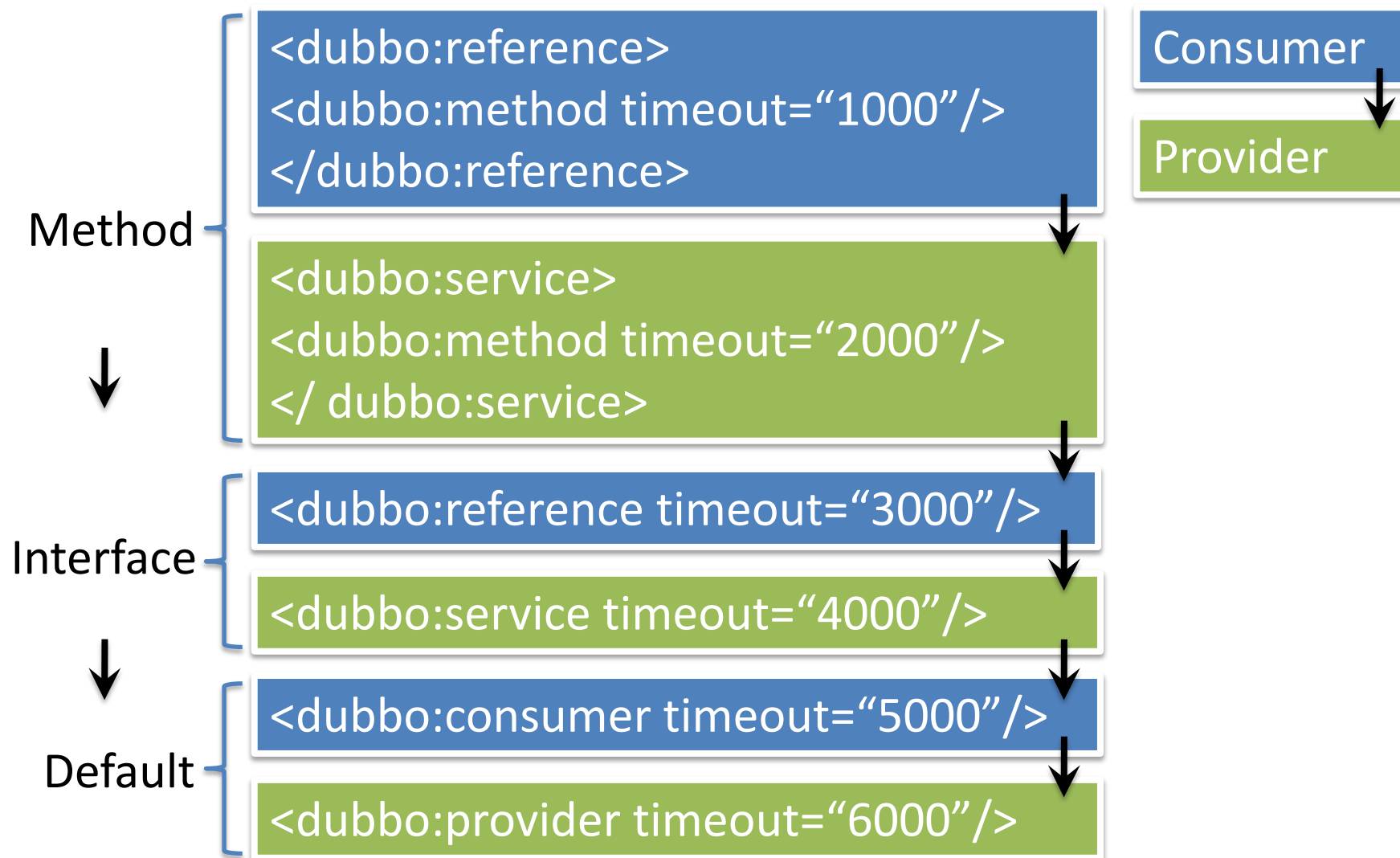


Config Type

- Service Identification
 - group
 - interface
 - version
- Service Governance
 - application
 - deprecated
 - ...
- Performance Optimize
 - timeout
 - threads
 - ...



Config Override





Config URL

Provider

- `dubbo://10.20.153.10:20880/xxxService?timeout=1000&xxx.timeout=1000&default.timeout=1000`

Registry

- `dubbo://10.20.153.10:20880/xxxService?timeout=2000&xxx.timeout=2000&default.timeout=2000`

Consumer

- `dubbo://10.20.153.10:20880/xxxService?timeout=3000&xxx.timeout=3000&default.timeout=3000`

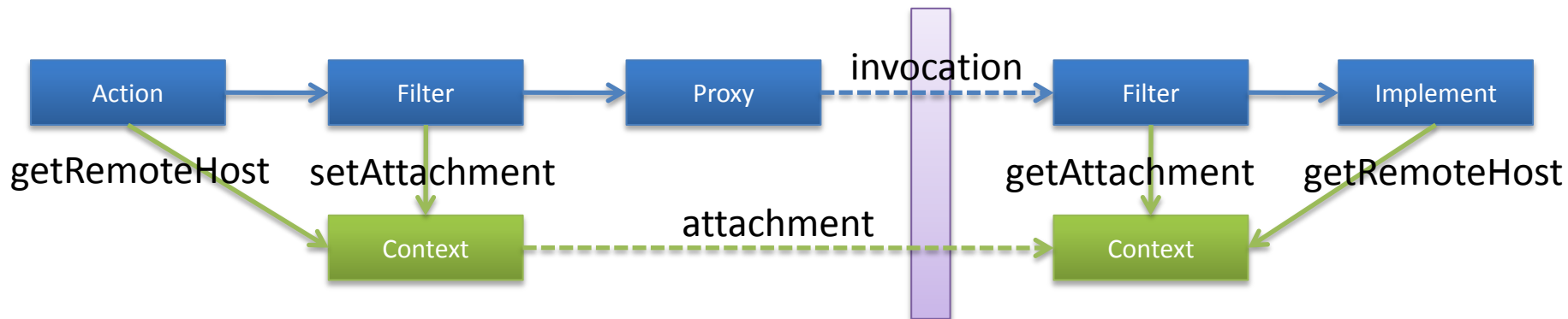


Programmable Config

- ServiceConfig
 - setApplication(new ApplicationConfig("xxx"));
 - setRegistry(new RegistryConfig("10.20.153.186"));
 - setInterface("com.xxx.XxxService");
 - export();
- ReferenceConfig
 - setApplication(new ApplicationConfig("yyy"));
 - setRegistry(new RegistryConfig("10.20.153.186"));
 - setInterface("com.xxx.XxxService");
 - get();



Context & Trace



- ThreadLocal Context

- `RpcContext.getContext().getRemoteHost();`
- `RpcContext.getContext().setAttachment("index", "1");`
- `RpcContext.getContext().getAttachment("index");`

- Like HTTP Cookie



Generic Service

- Java interface

```
MemberService memberService = (MemberService) applicationContext.getBean("memberService");  
memberService.findPerson(memberId);
```

- Generic interface

```
GenericService memberService = applicationContext.getBean("memberService");  
memberService.invoke("findPerson", new String[] {"java.lang.String"}, new Object[] {memberId});
```

- Java implementation

```
public class MemberSericeImpl implements MemberService {  
}
```

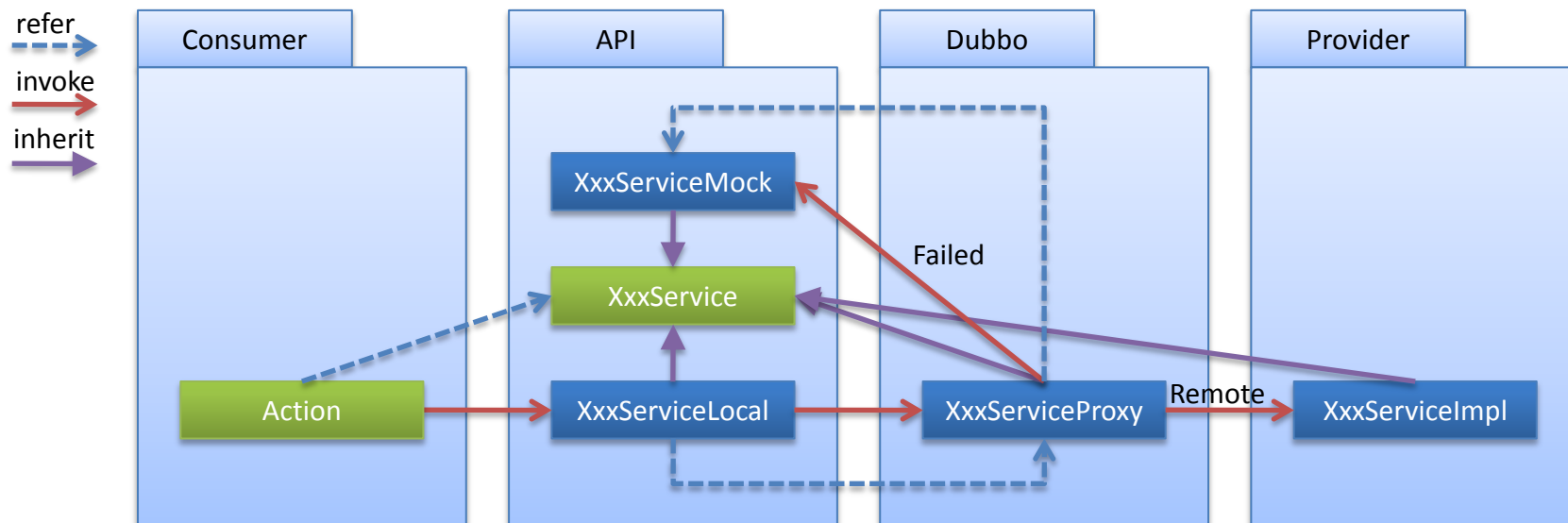
- Generic implementation

```
public class MemberSericeImpl implements GenericService {  
    public Object invoke(String method, String[] ptypes, Object[] args) {  
    }  
}
```





Local & Mock



- `dubbo://10.20.153.10?local=true&mock=true`
- Constructor:
 - `public XxxServiceLocal(XxxService xxxService) {}`
 - `public XxxServiceMock() {}`



Summary

Framework

- Layers & Modules



SPI

- Microkernel & SPI
- Protocol & ProxyFactory & Filter
- Cluster & Directory & Router & LoadBalance
- Transporter & Serialization & ThreadPool
- TelnetHandler & StatusChecker



API

- Config & URL
- Context & Generic & Local



Resources

- Documents:
 - <http://b2b-doc.alibaba-inc.com/display/RC/Dubbo>
 - http://b2b-doc.alibaba-inc.com/display/RC/Dubbo_SPI_Reference
 - http://b2b-doc.alibaba-inc.com/display/RC/Dubbo_API_Reference
 - http://b2b-doc.alibaba-inc.com/display/RC/Dubbo_Contribute
- Supports:
 - 梁飞 (william.liangf@alibaba-inc.com)
 - 刘超(chao.liuc@alibaba-inc.com)
 - 李鼎(ding.lid@alibaba-inc.com)
 - 刘昊旻(haomin.liuhm@alibaba-inc.com)
 - 陈雷(tony.chenl@alibaba-inc.com)



Q & A

- Q & A