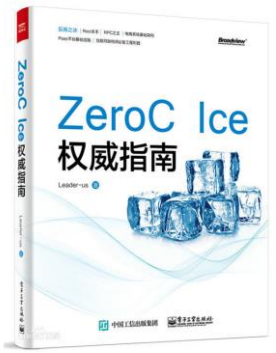
分布式高端架构系列

Zeroc ICE

——微服务架构之王

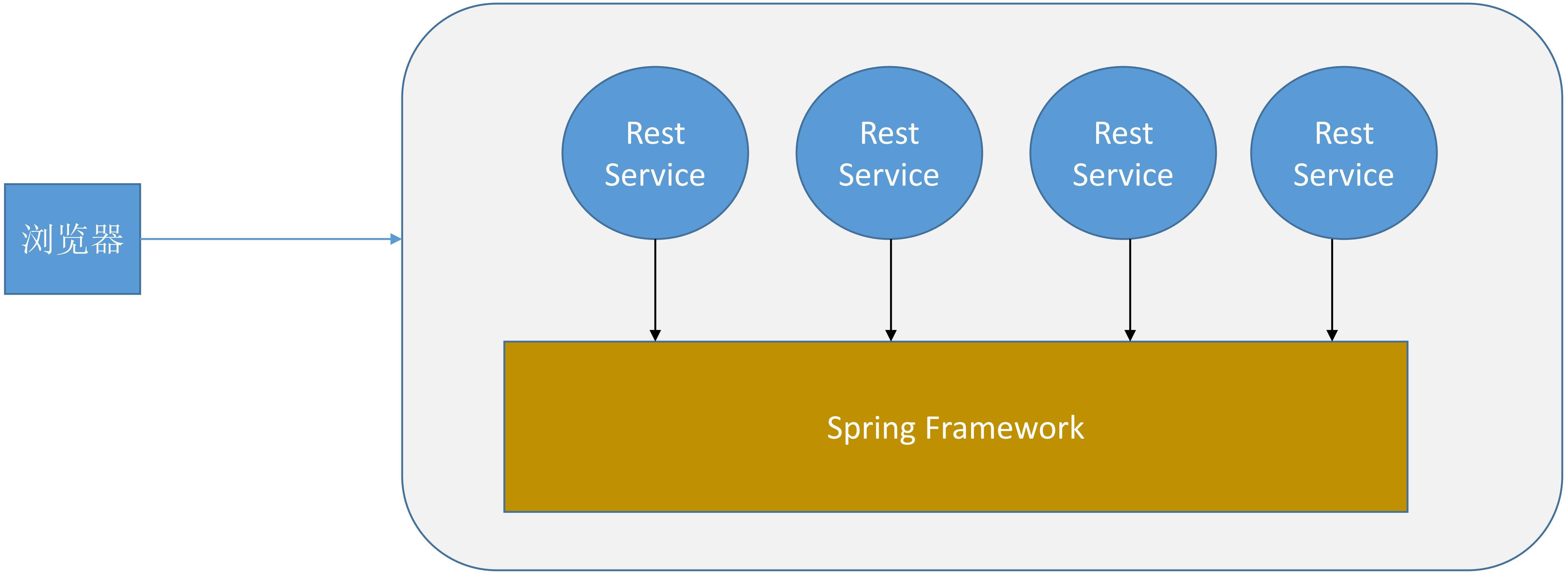


Leader-us@Mycat

一：微服务架构概述

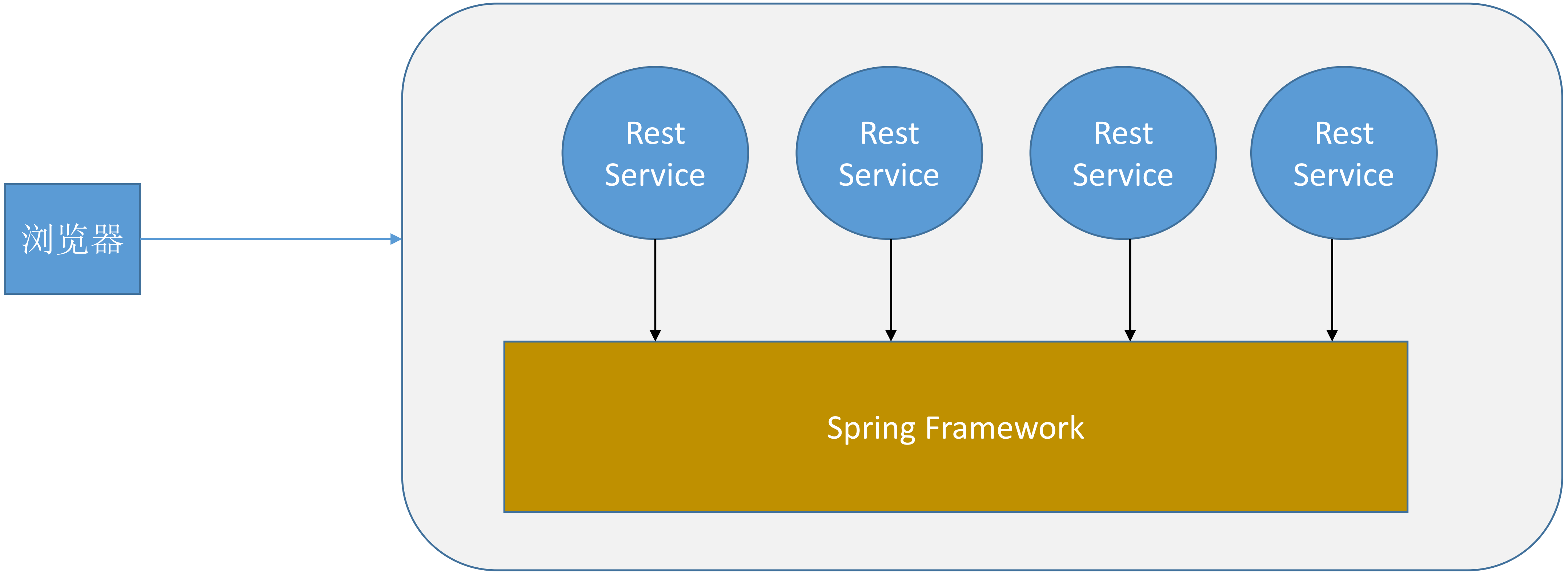
1：什么是单体应用

主要业务逻辑都运行在**一个进程**里的程序



1：什么是单体应用

主要业务逻辑都运行在**一个进程**里的程序



2:单体应用的七宗罪

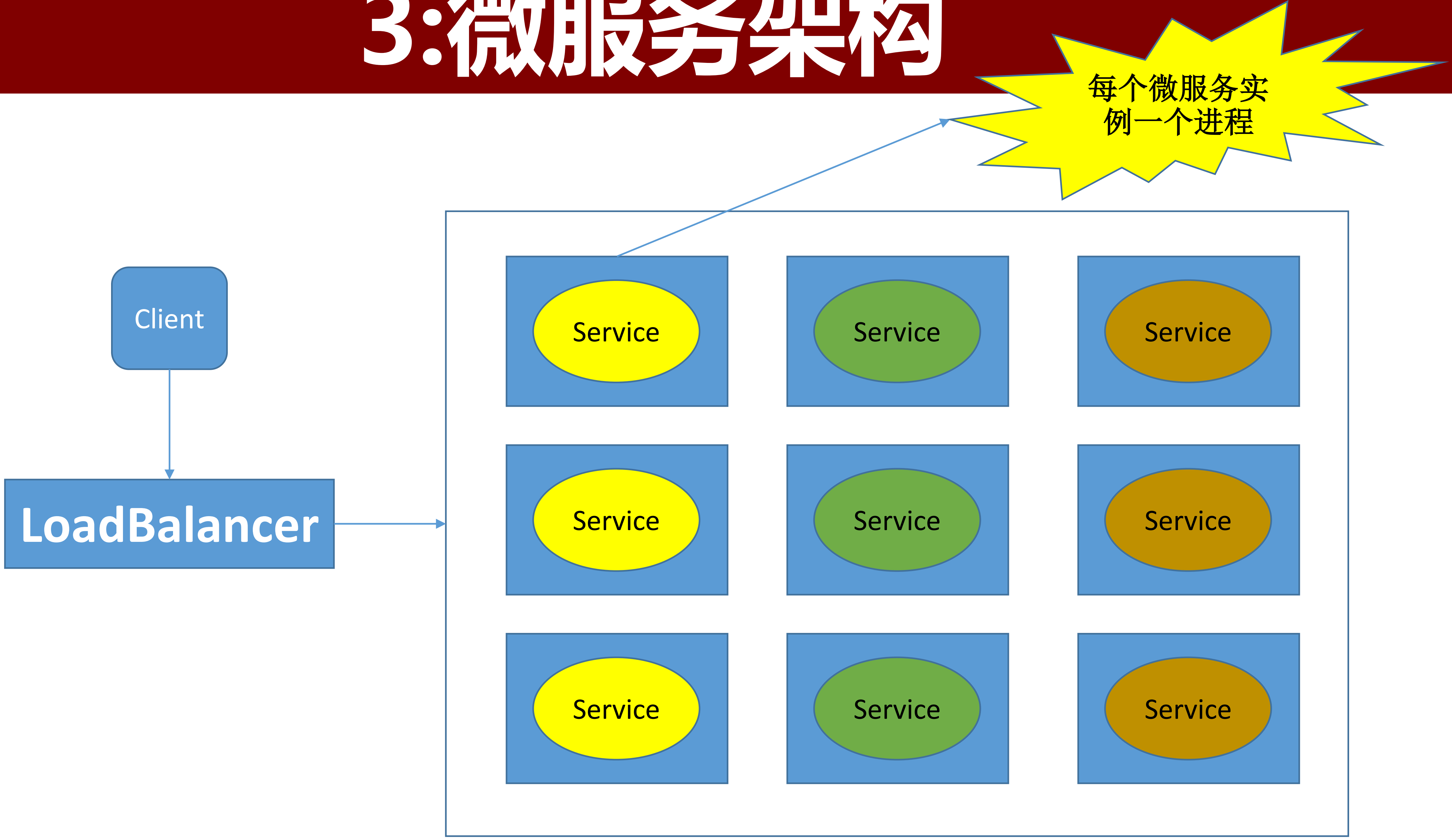
No1:先天性缺陷

No2:系统性风险

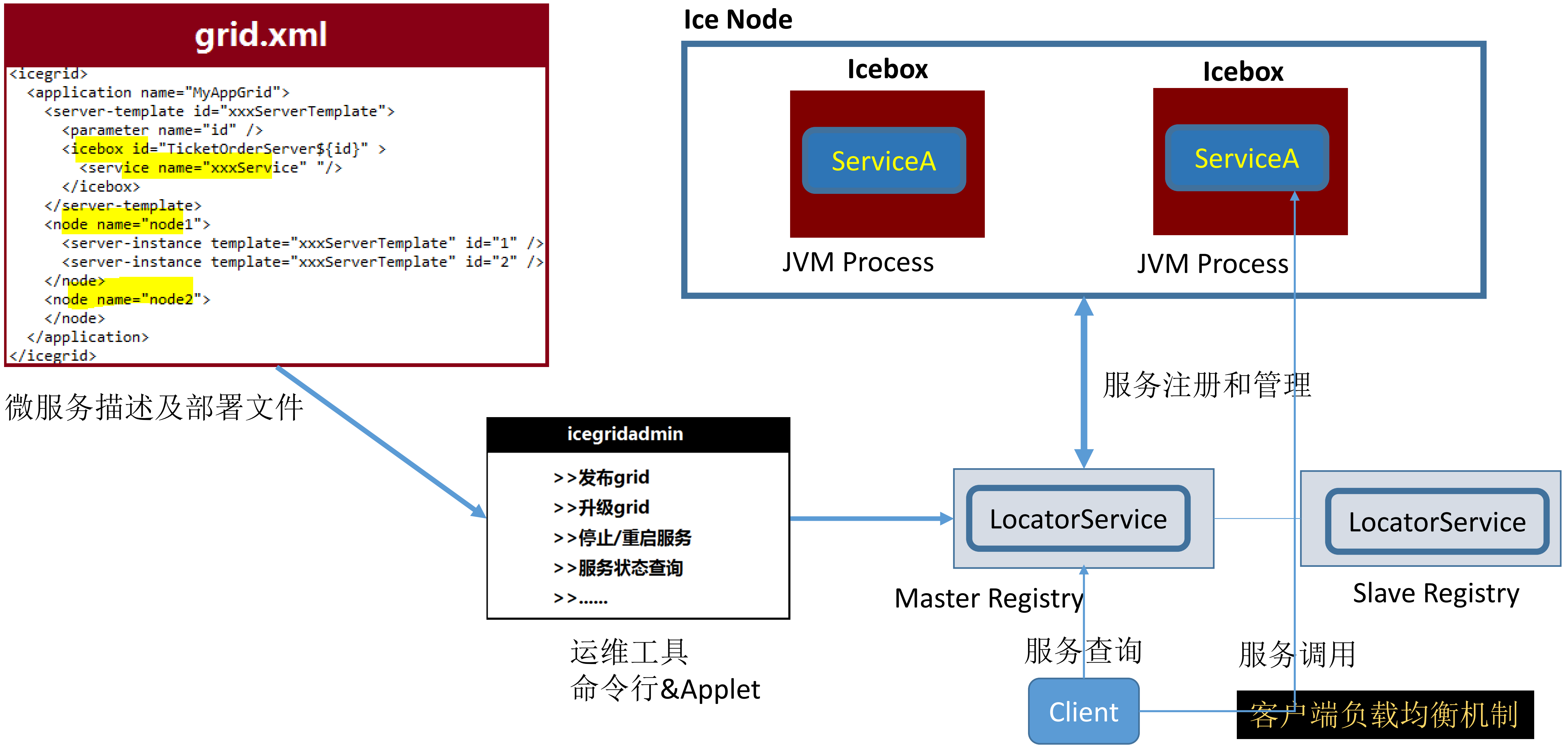
No3:运维风险

No4:难以可持续发展

3:微服务架构


每个微服务实例一个进程
Client
LoadBalancer
Service
Service
Service
Service
Service
Service
Service
Service
Service

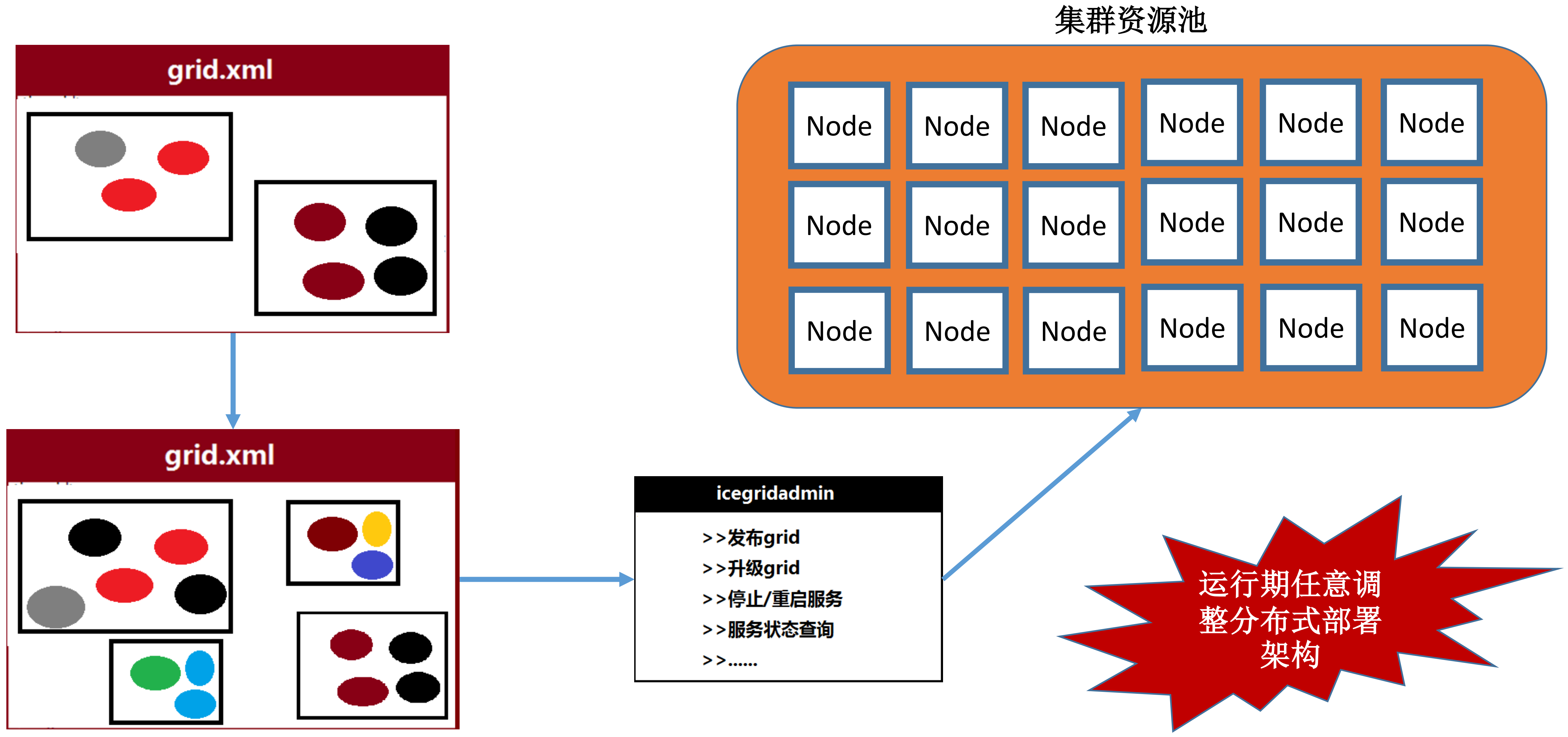
4：Ice微服务架构



grid.xml

```
<icegrid>
  <application name="MyAppGrid">
    <server-template id="xxxServerTemplate">
      <parameter name="id" />
      <icebox id="TicketOrderServer${id}" >
        <service name="xxxService" "/>
      </icebox>
    </server-template>
    <node name="node1">
      <server-instance template="xxxServerTemplate" id="1" />
      <server-instance template="xxxServerTemplate" id="2" />
    </node>
    <node name="node2">
    </node>
  </application>
</icegrid>
```

4：Ice微服务架构特点



5：微服务架构的优点

No1:先天分布式

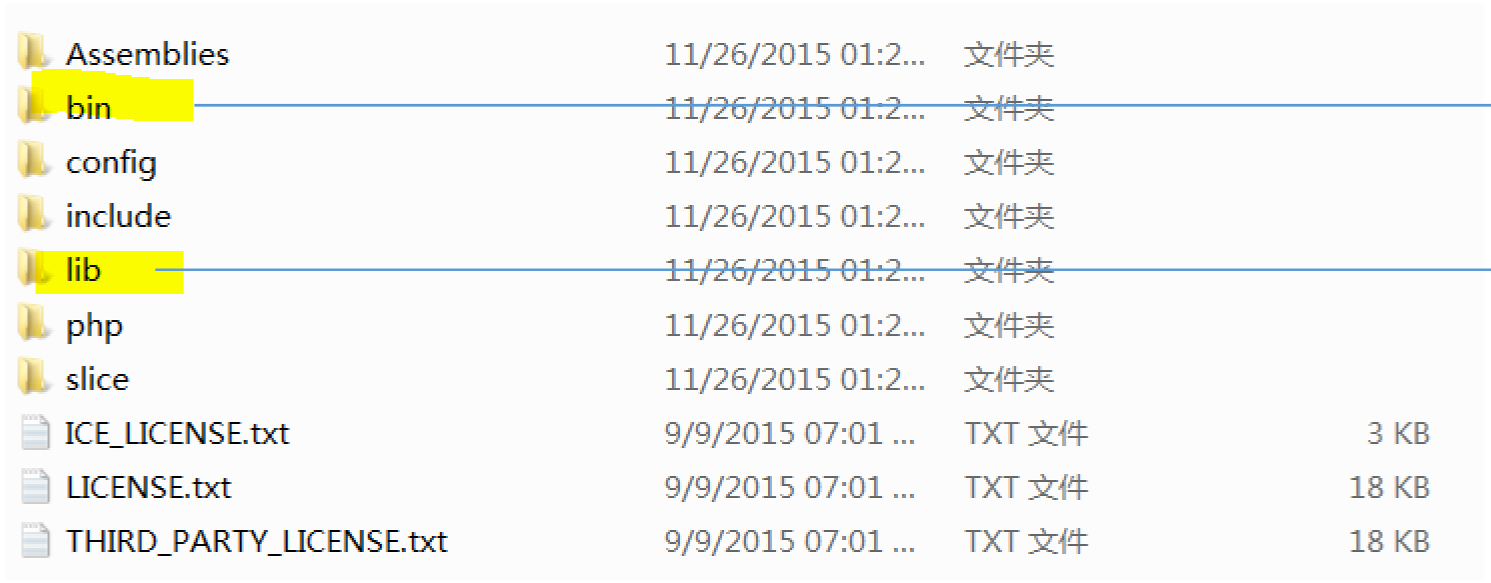
No2:无状态（尽量）

No3:积木式发展

二：Zeroc Ice安装

1：安装ICE SDK环境

官方：https://zeroc.com/，**区分平台**，安装包比较大, 是因为各个版本的库文件都包括了



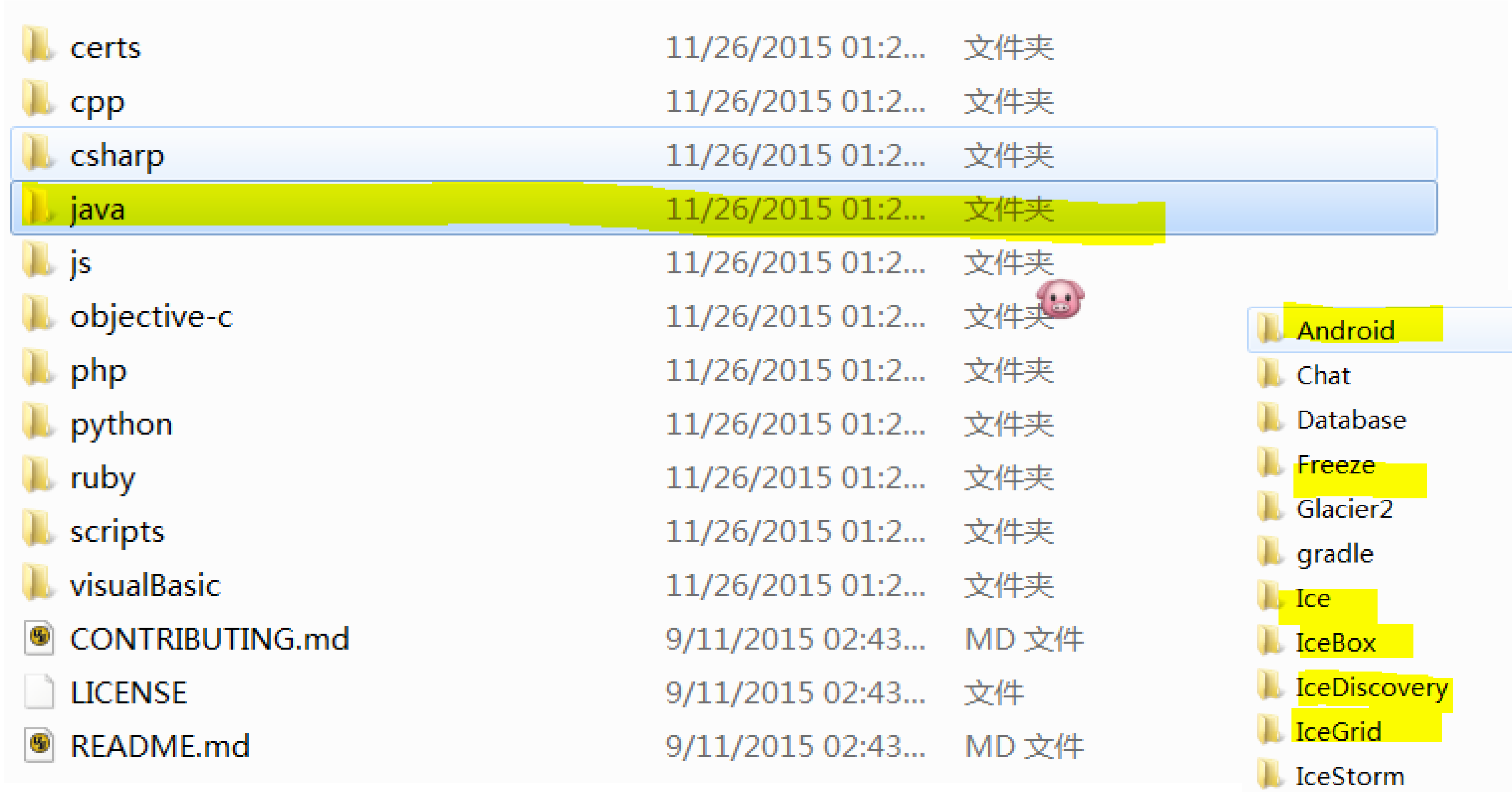
Assemblies	11/26/2015 01:2...	文件夹	
bin	11/26/2015 01:2...	文件夹	
config	11/26/2015 01:2...	文件夹	
include	11/26/2015 01:2...	文件夹	
lib	11/26/2015 01:2...	文件夹	
php	11/26/2015 01:2...	文件夹	
slice	11/26/2015 01:2...	文件夹	
ICE_LICENSE.txt	9/9/2015 07:01 ...	TXT 文件	3 KB
LICENSE.txt	9/9/2015 07:01 ...	TXT 文件	18 KB
THIRD_PARTY_LICENSE.txt	9/9/2015 07:01 ...	TXT 文件	18 KB

bin → 命令行工具，如slice2java Icegridnode, icegridadmin

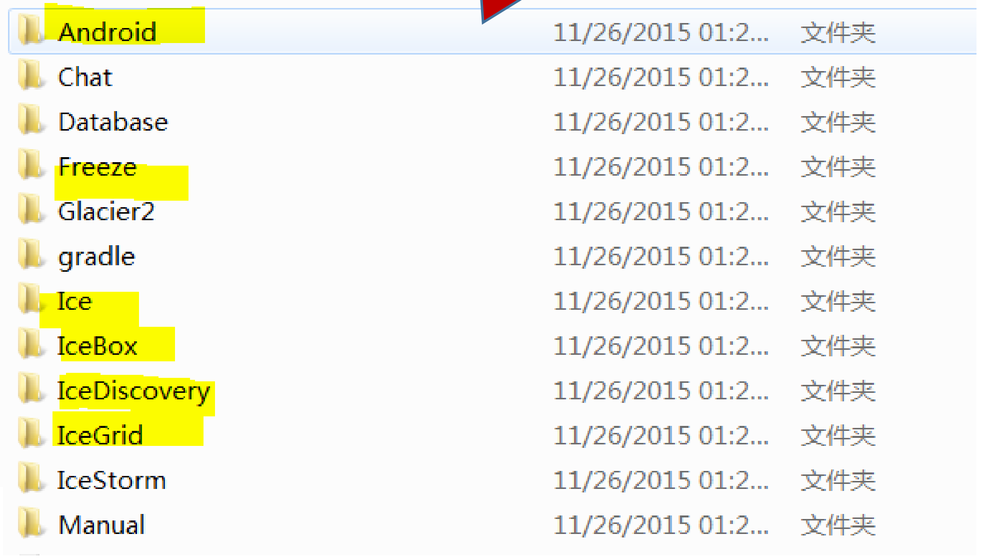
lib → Ice.jar,icebox.jar等文件

2：安装Ice demos

Ice-3.6.1-demos部分包括各个语言的例子，建议安装的时候选择安装



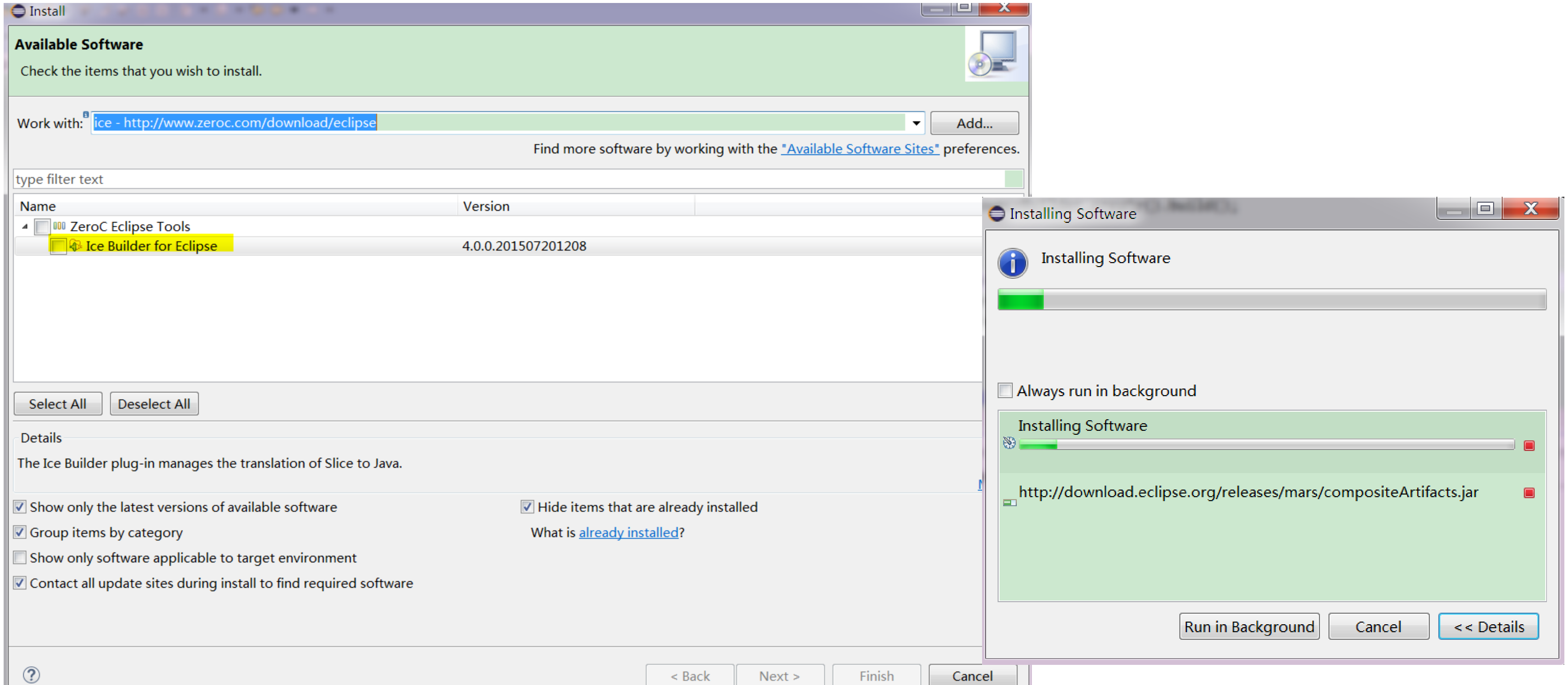
certs	11/26/2015 01:2...	文件夹
cpp	11/26/2015 01:2...	文件夹
csharp	11/26/2015 01:2...	文件夹
java	11/26/2015 01:2...	文件夹
js	11/26/2015 01:2...	文件夹
objective-c	11/26/2015 01:2...	文件夹
php	11/26/2015 01:2...	文件夹
python	11/26/2015 01:2...	文件夹
ruby	11/26/2015 01:2...	文件夹
scripts	11/26/2015 01:2...	文件夹
visualBasic	11/26/2015 01:2...	文件夹
CONTRIBUTING.md	9/11/2015 02:43...	MD 文件
LICENSE	9/11/2015 02:43...	文件
README.md	9/11/2015 02:43...	MD 文件



Android	11/26/2015 01:2...	文件夹
Chat	11/26/2015 01:2...	文件夹
Database	11/26/2015 01:2...	文件夹
Freeze	11/26/2015 01:2...	文件夹
Glacier2	11/26/2015 01:2...	文件夹
gradle	11/26/2015 01:2...	文件夹
Ice	11/26/2015 01:2...	文件夹
IceBox	11/26/2015 01:2...	文件夹
IceDiscovery	11/26/2015 01:2...	文件夹
IceGrid	11/26/2015 01:2...	文件夹
IceStorm	11/26/2015 01:2...	文件夹
Manual	11/26/2015 01:2...	文件夹

3：安装Eclipse Ice插件

插件用于自动完成slice接口定义到java 接口包的源码生成
插件地址http://www.zeroc.com/download/eclipse



Install

Available Software

Check the items that you wish to install.

Work with: ice - http://www.zeroc.com/download/eclipse

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
ZeroC Eclipse Tools	
Ice Builder for Eclipse	4.0.0.201507201208

Details

The Ice Builder plug-in manages the translation of Slice to Java.

- [✓] Show only the latest versions of available software
- [✓] Group items by category
- [ ] Show only software applicable to target environment
- [✓] Contact all update sites during install to find required software
- [✓] Hide items that are already installed

What is already installed?

Installing Software

Installing Software

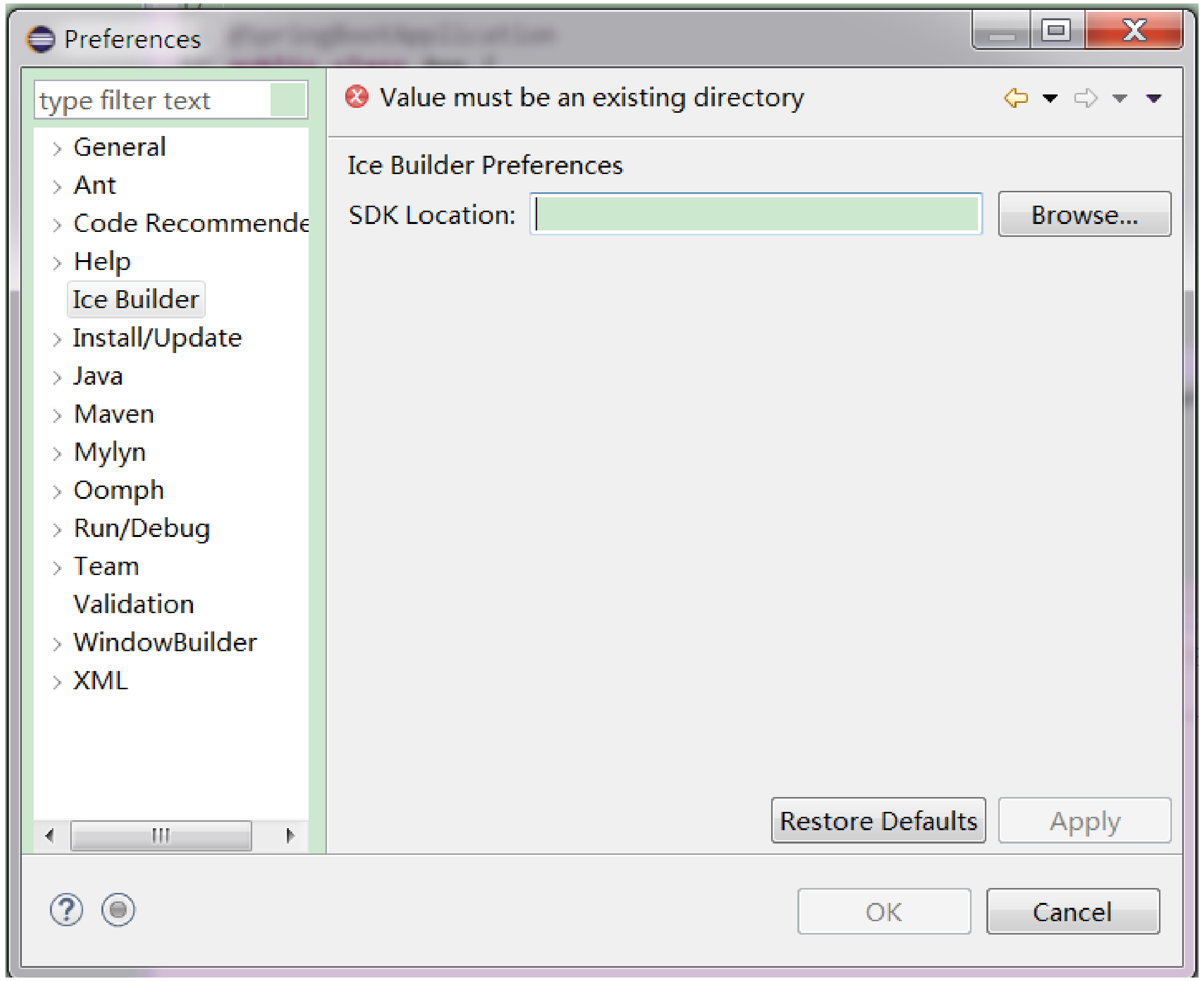
[ ] Always run in background

Installing Software

http://download.eclipse.org/releases/mars/compositeArtifacts.jar

4：Ice插件设置

Ice Builder插件需要设置Ice SDK的路径，eclipse->preferences菜单里进行设置



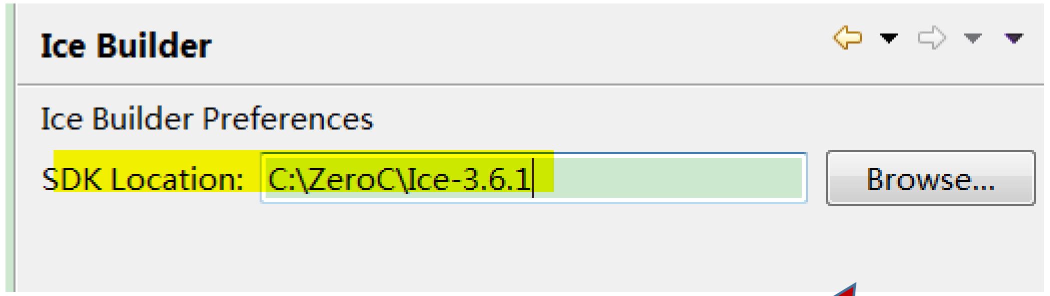
Preferences

type filter text

Value must be an existing directory

Ice Builder Preferences

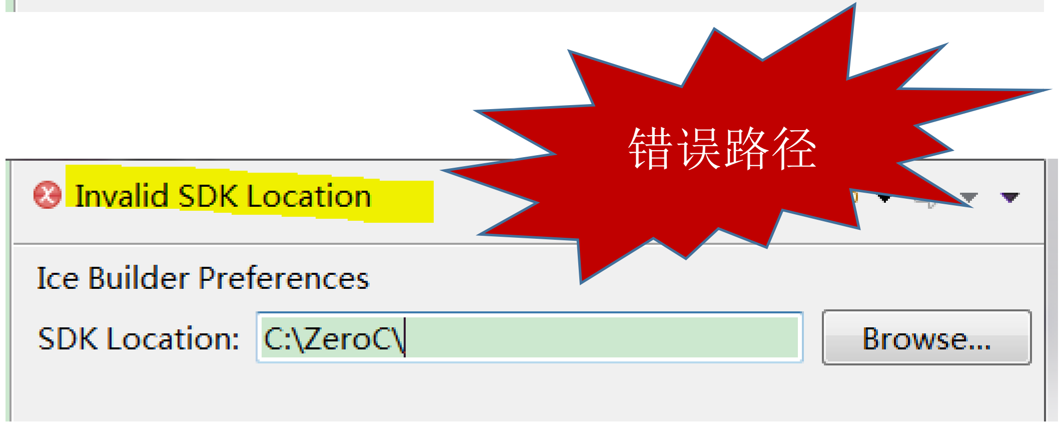
SDK Location:



Ice Builder

Ice Builder Preferences

SDK Location: C:\ZeroC\Ice-3.6.1



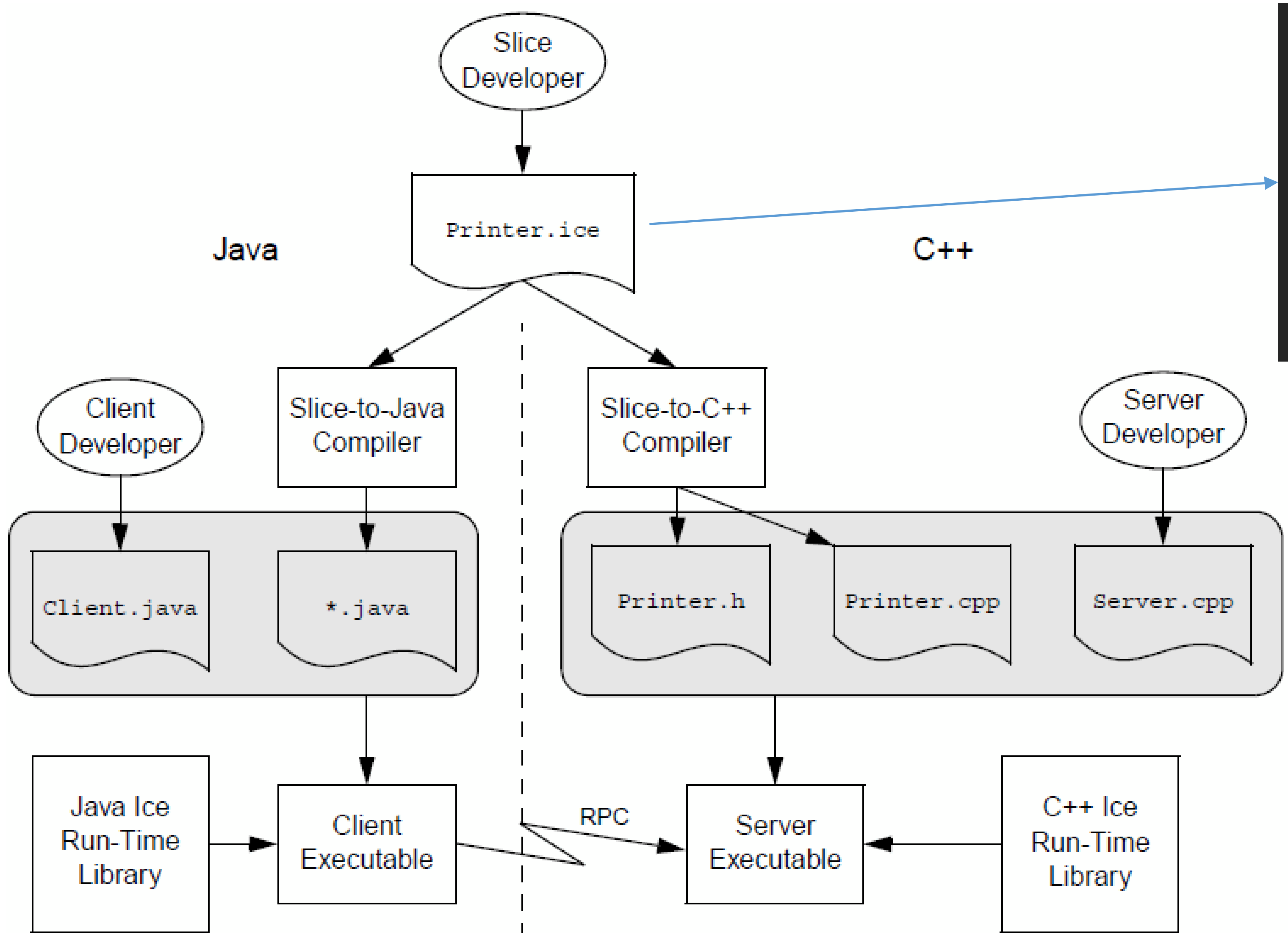
Invalid SDK Location

Ice Builder Preferences

SDK Location: C:\ZeroC\

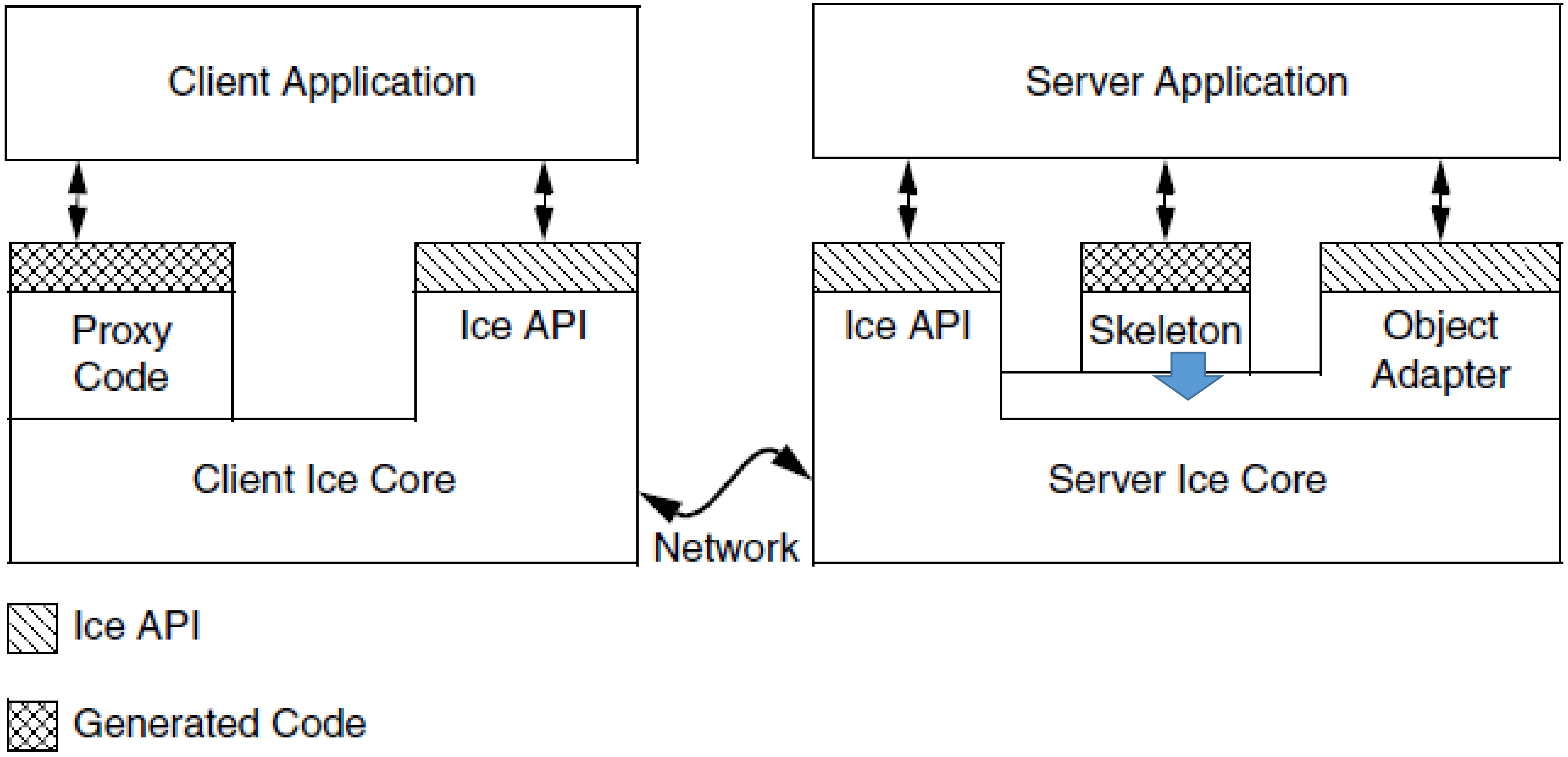
三：Zeroc Ice入门

1：服务定义&Slice



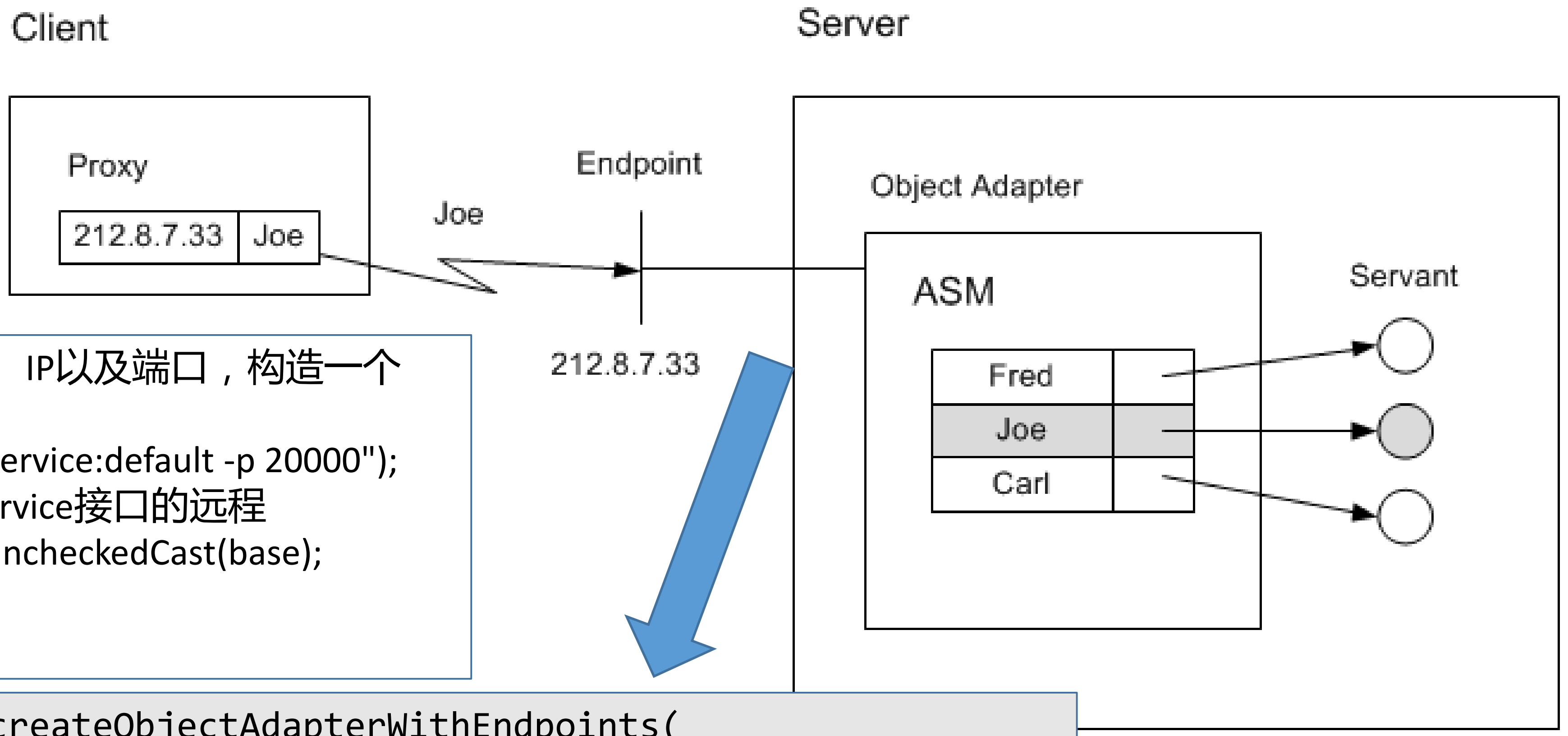
```
module demo{
interface MyService{
string hellow();
};
};
```

2：ＩＣＥ通信原理


Client Application
Proxy Code
Ice API
Client Ice Core
Network
Server Application
Ice API
Skeleton
Object Adapter
Server Ice Core
Ice API
Generated Code

3 ：基本概念

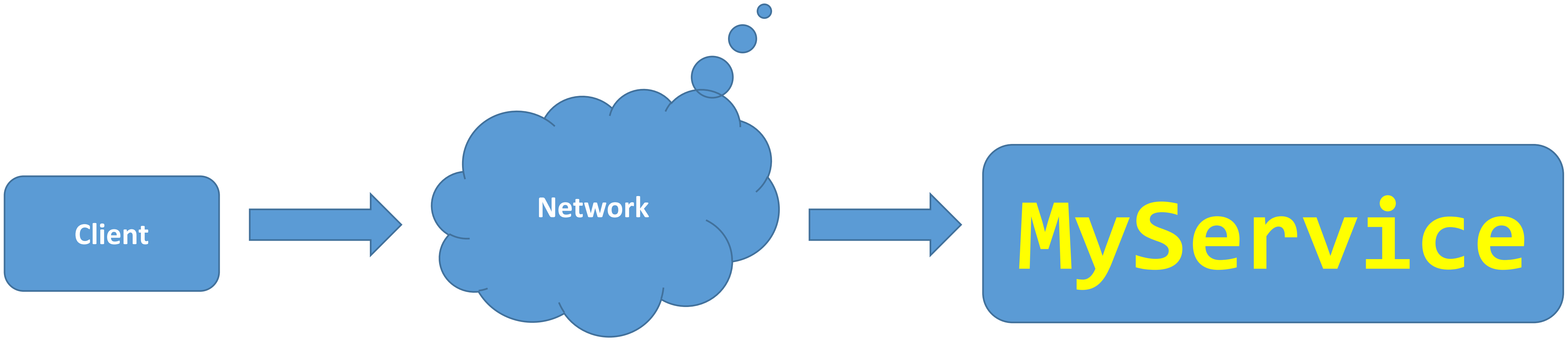
An Ice object is an abstraction. Ice objects do not physically The concept of an Ice object is made real by a servant



```
// 传入远程服务单元的名称、网络协议、IP以及端口，构造一个Proxy对象
Ice.ObjectPrx base = ic.stringToProxy("MyService:default -p 20000");
// 通过checkedCast向下转型，获取MyService接口的远程
MyServicePrx prxy = MyServicePrxHelper.uncheckedCast(base);
// 调用服务方法
String rt=prxy.hellow();
```

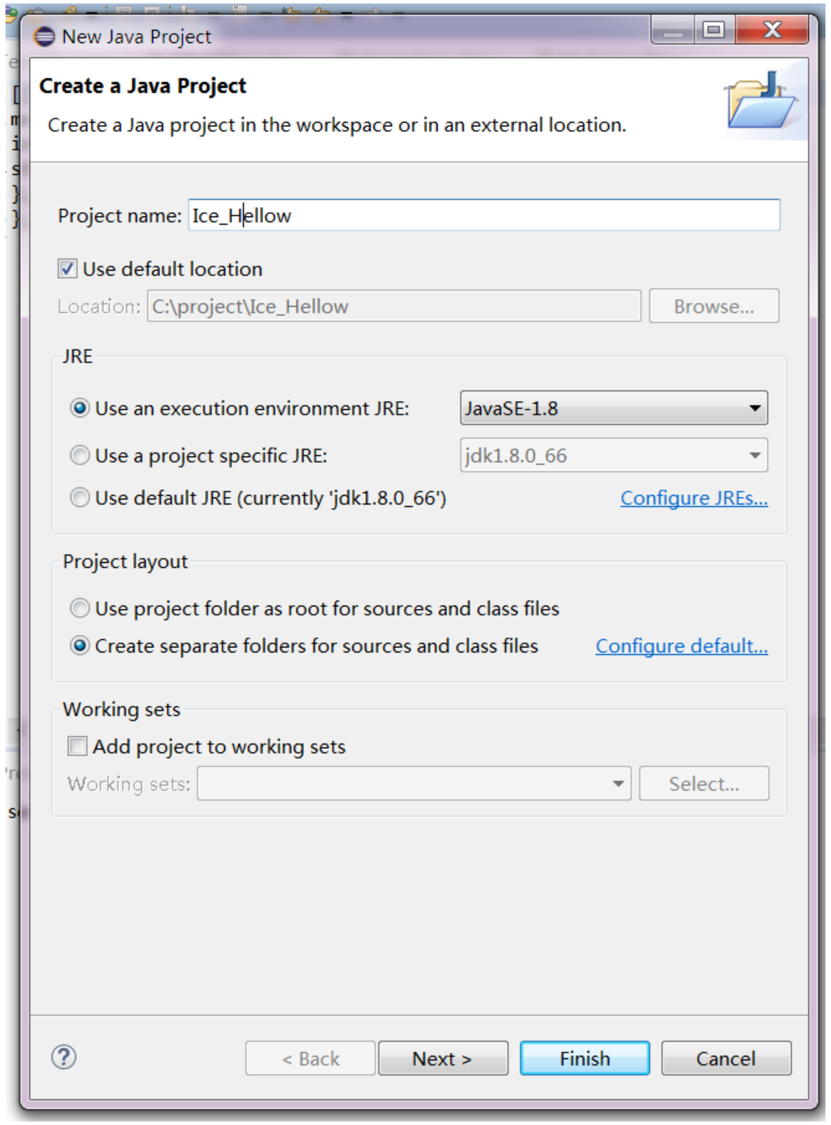
```
Ice.ObjectAdapter adapter = ic.createObjectAdapterWithEndpoints(
"MyServiceAdapter", "default -p 20000");
// 实例化一个MyService服务对象(Servant)
MyServiceImpl servant = new MyServiceImpl();
// 将Servant增加到ObjectAdapter中，并将Servant关联到ID为MyService的Ice Object
adapter.add(servant, Ice.Util.stringToIdentity("MyService"));
// 激活ObjectAdapter
adapter.activate();
```

4 : Hello World(1)



```
string hellow()
```

4：Hello World(2)



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: Ice_Hellow

[✓] Use default location

Location: C:\project\Ice_Hellow   Browse...

JRE

[● Use an execution environment JRE:] JavaSE-1.8

○ Use a project specific JRE: jdk1.8.0_66

○ Use default JRE (currently 'jdk1.8.0_66')   Configure JREs...

Project layout

○ Use project folder as root for sources and class files

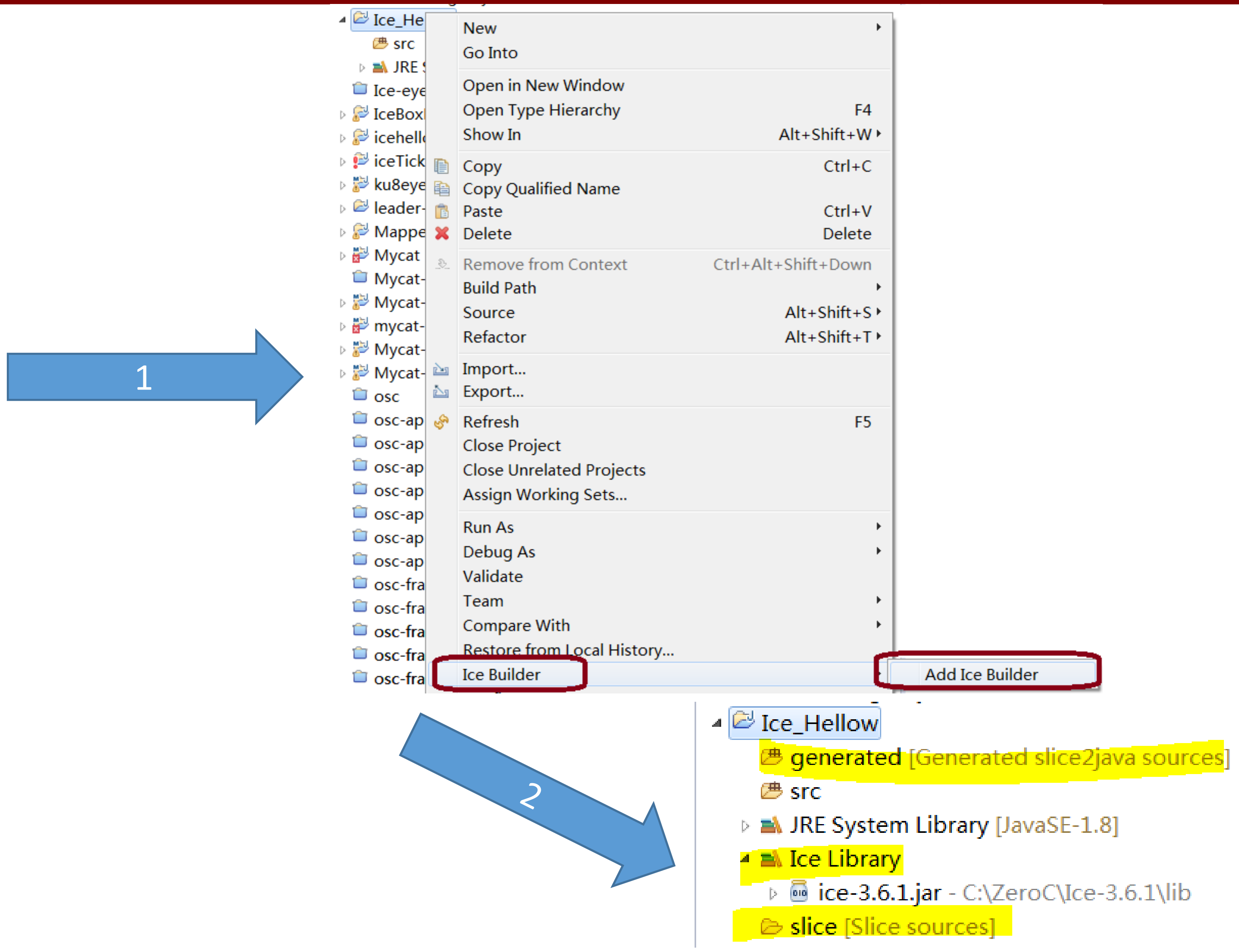
[● Create separate folders for sources and class files]   Configure default...

Working sets

[ ] Add project to working sets

Working sets: Select...

< Back   Next >   Finish   Cancel



4：Hello World(3)

MyService服务接口定义

myservice.ice

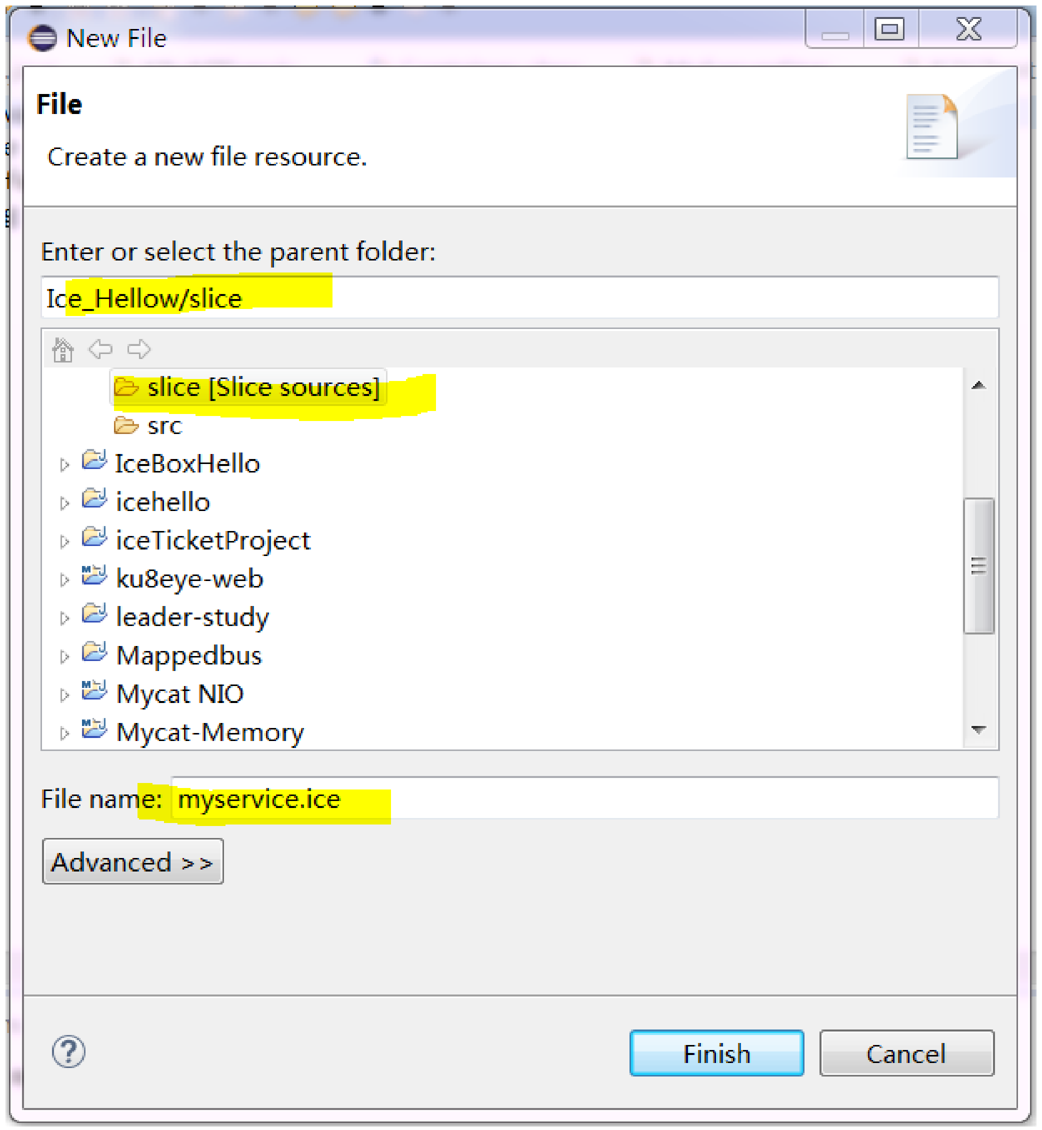
映射为Java包名

```
[["java:package:com.my"]]
module demo{
interface MyService{
string hellow();
};
};
```

模块名

服务接口

4：Hello World(4)



New File

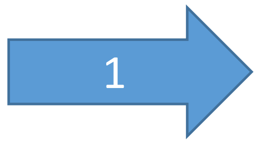
File

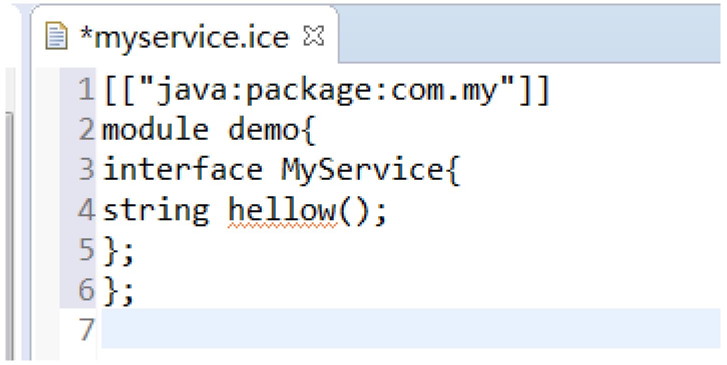
Create a new file resource.

Enter or select the parent folder:

Ice_Hellow/slice

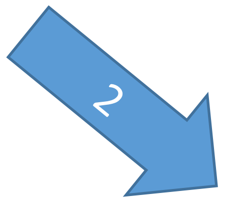
File name: myservice.ice

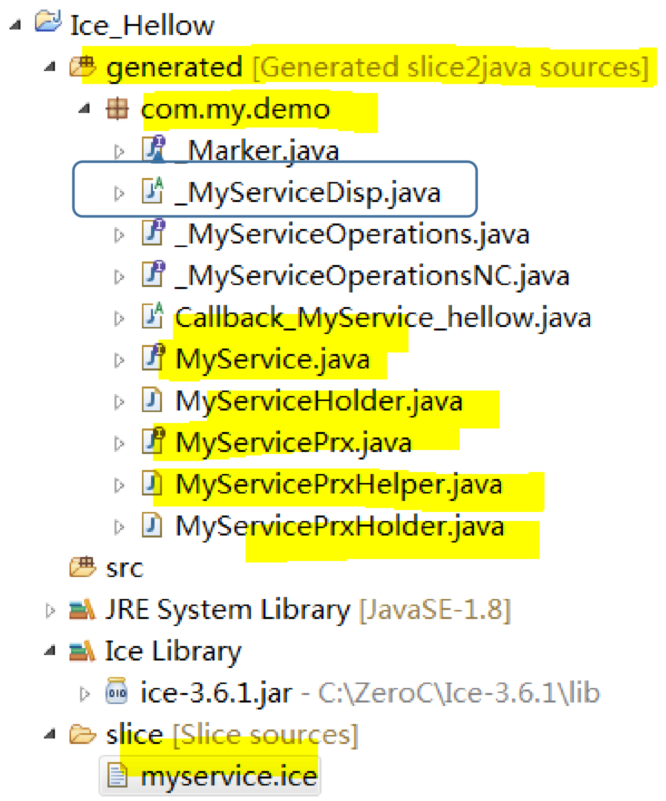




*myservice.ice

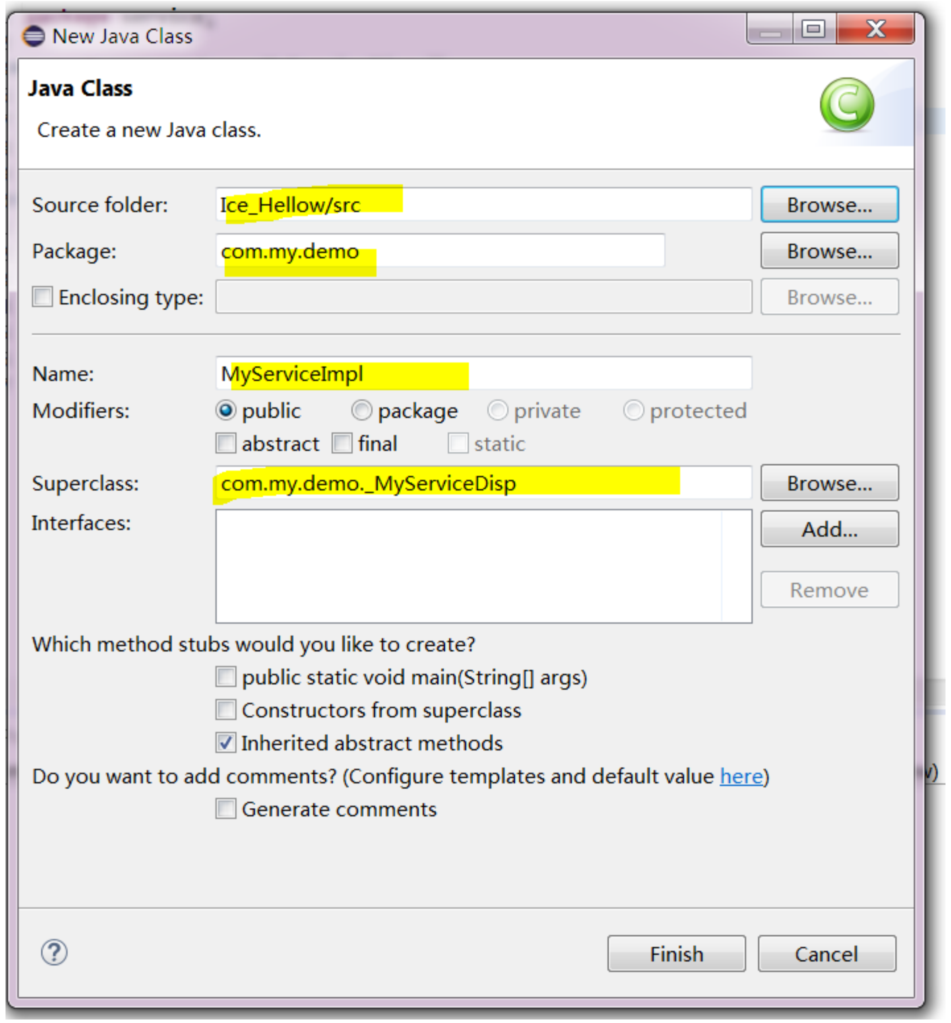
```
[["java:package:com.my"]]
module demo{
interface MyService{
string hellow();
};
};

```





4：Hello World(5)

MyService服务端开发



New Java Class

Java Class

Create a new Java class.

Source folder: Ice_Hellow/src
Package: com.my.demo
☐ Enclosing type:

Name: MyServiceImpl
Modifiers: [✓ public] package private protected
☐ abstract ☐ final ☐ static

Superclass: com.my.demo._MyServiceDisp
Interfaces:

Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)
☐ Generate comments

1

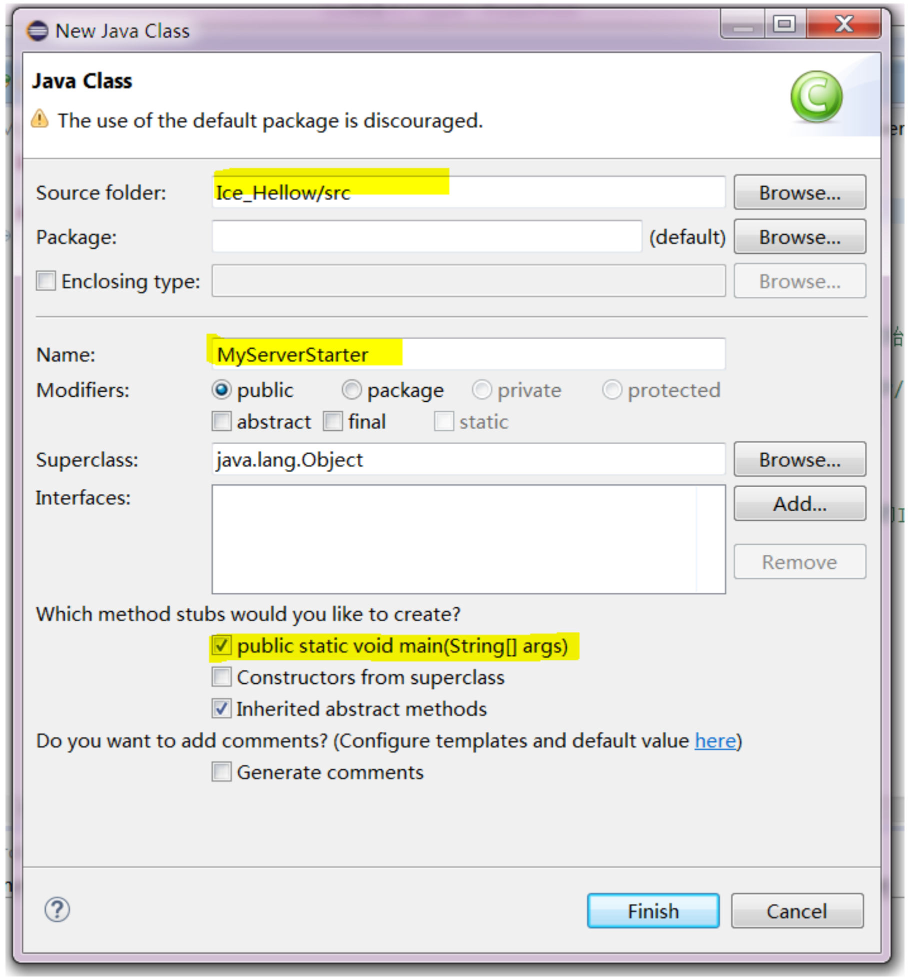
```
package com.my.demo;
import Ice.Current;
public class MyServiceImpl extends
_MyServiceDisp {

@Override
public String hellow(Current __current) {
// TODO Auto-generated method stub
return null;
}
}
```

2

```
@Override
public String hellow(Current __current) {
return "Hello world";

}
```

4：Hello World(6)

Server Starter程序



New Java Class

Java Class

The use of the default package is discouraged.

Source folder: Ice_Hellow/src

Package: (default)

Enclosing type:

Name: MyServerStarter

Modifiers: [✓ public] package private protected

abstract final static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

[✓] public static void main(String[] args)

[ ] Constructors from superclass

[✓] Inherited abstract methods

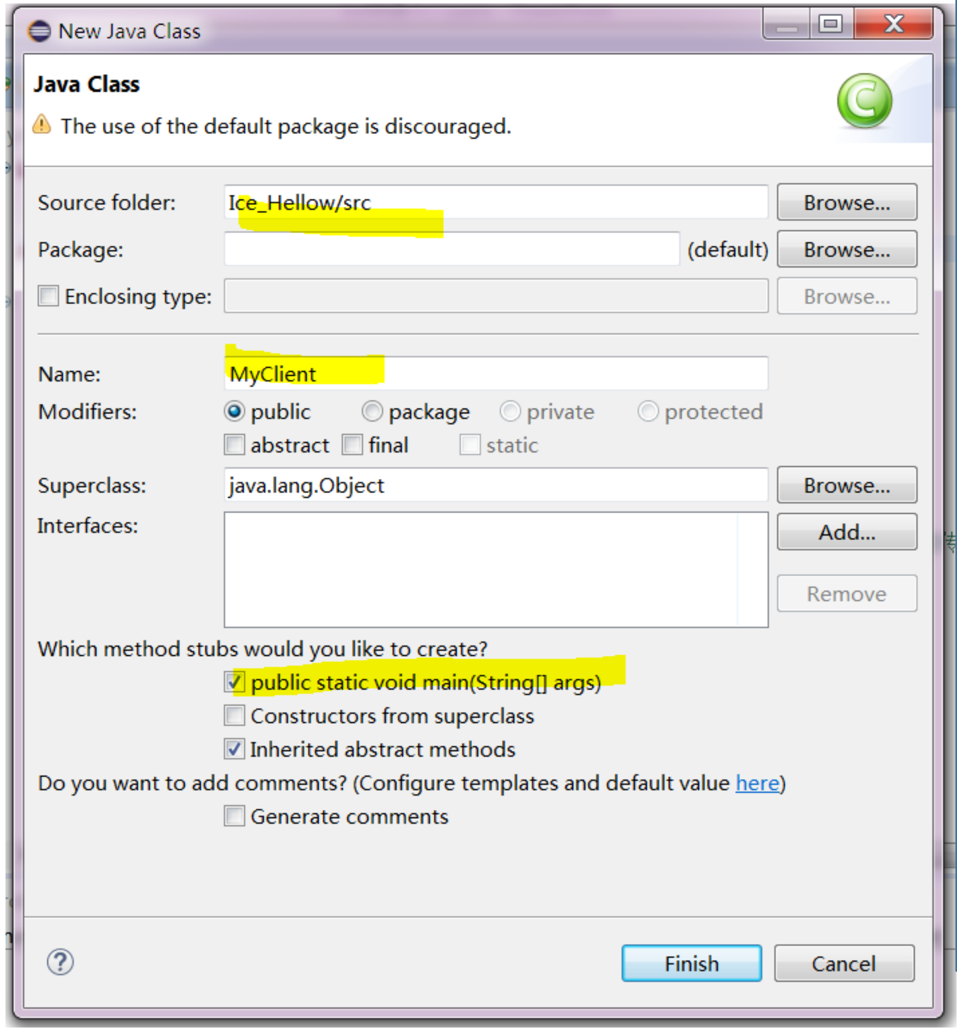
Do you want to add comments? (Configure templates and default value here)

[ ] Generate comments

```
public static void main(String[] args) {
int status = 0;
Ice.Communicator ic = null;
try {
// 初始化Communicator对象，args可以传一些初始化参数，如连接超时，初始化客户端连接池的数量等
ic = Ice.Util.initialize(args);
// 创建名为MyServiceAdapter的ObjectAdapter，使用缺省的通信协议（TCP/IP 端口为20000的请求）
Ice.ObjectAdapter adapter = ic.createObjectAdapterWithEndpoints(
"MyServiceAdapter", "default -p 20000");
// 实例化一个MyService服务对象(Servant)
MyServiceImpl servant = new MyServiceImpl();
// 将Servant增加到ObjectAdapter中，并将Servant关联到ID为MyService的Ice Object
adapter.add(servant, Ice.Util.stringToIdentity("MyService"));
// 激活ObjectAdapter
adapter.activate();
// 让服务在退出之前，一直持续对请求的监听
System.out.print("server started ");
ic.waitForShutdown();
} catch (Exception e) {
e.printStackTrace();
status = 1;
} finally {
if (ic != null) {
ic.destroy();
}
}
System.exit(status);
}
```

4：Hello World(7)

Client程序


New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder:	Ice_Hellow/src	Browse...
Package:	(default)	Browse...
☐ Enclosing type:		Browse...
Name:	MyClient	
Modifiers:	[● public] ○ package ○ private ○ protected ☐ abstract ☐ final ☐ static	
Superclass:	java.lang.Object	Browse...
Interfaces:		Add... Remove

Which method stubs would you like to create?

- ☑ public static void main(String[] args)
- ☐ Constructors from superclass
- ☑ Inherited abstract methods

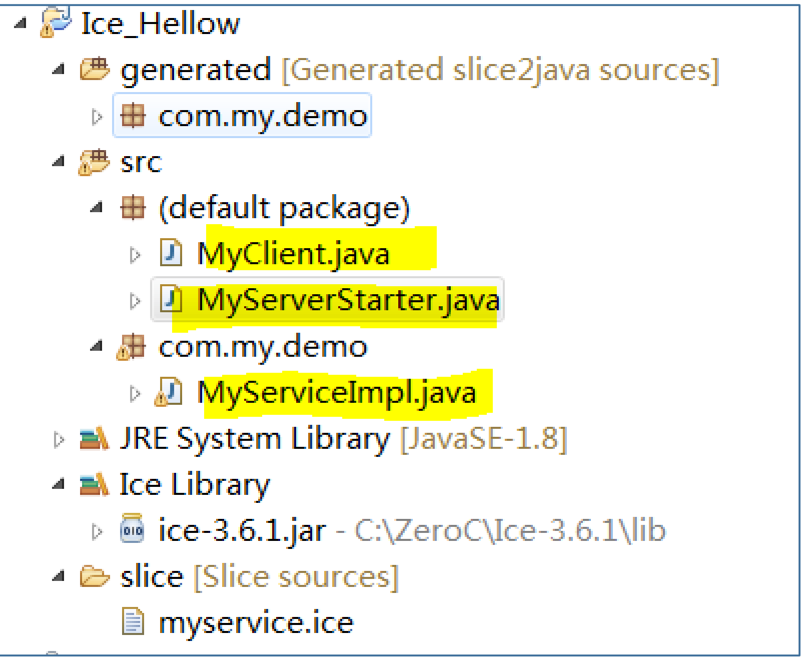
Do you want to add comments? (Configure templates and default value here)

- ☐ Generate comments

```
public static void main(String[] args) {
int status = 0;
Ice.Communicator ic = null;
try {
// 初始化通信器
ic = Ice.Util.initialize(args);
// 传入远程服务单元的名称、网络协议、IP以及端口，构造一个Proxy对象
Ice.ObjectPrx base = ic
.stringToProxy("MyService:default -p 20000");
// 通过checkedCast向下转型，获取MyService接口的远程，并同时检测根据传入的名称获取服务单元是
否OnlineBook的代理接口，如果不是则返回null对象
MyServicePrx prxy = MyServicePrxHelper.uncheckedCast(base);
if (prxy == null) {
throw new Error("Invalid proxy");
}
// 调用服务方法
String rt=prxy.hellow();
System.out.print(rt);
} catch (Exception e) {
e.printStackTrace();
status = 1;
} finally {
if (ic != null) {
ic.destroy();
}
}
System.exit(status);
}
```

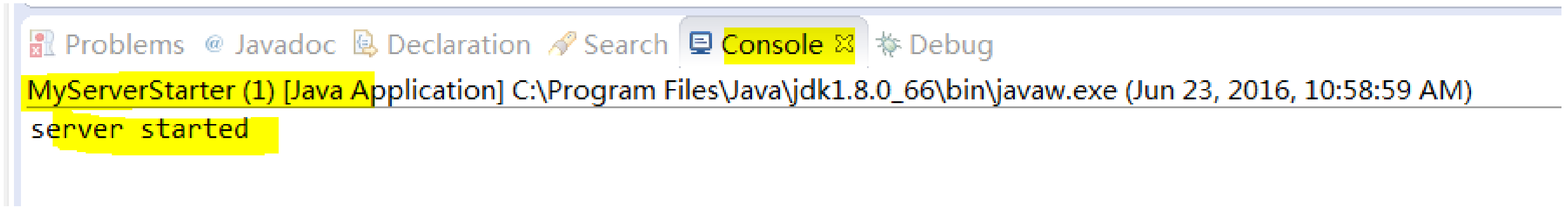
4：Hello World(8)

完工测试



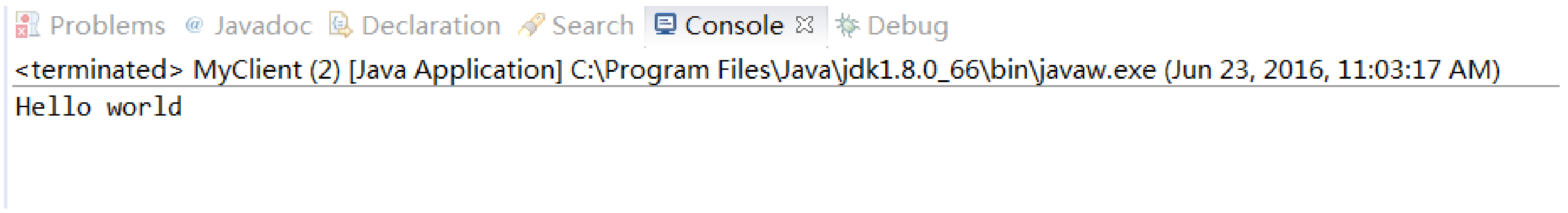
Ice_Hellow
- generated [Generated slice2java sources]
  - com.my.demo
- src
  - (default package)
    - MyClient.java
    - MyServerStarter.java
  - com.my.demo
    - MyServiceImpl.java
- JRE System Library [JavaSE-1.8]
- Ice Library
  - ice-3.6.1.jar - C:\ZeroC\Ice-3.6.1\lib
- slice [Slice sources]
  - myservice.ice

一：启动Server



```
MyServerStarter (1) [Java Application] C:\Program Files\Java\jdk1.8.0_66\bin\javaw.exe (Jun 23, 2016, 10:58:59 AM)
server started
```

二：启动Client



```
<terminated> MyClient (2) [Java Application] C:\Program Files\Java\jdk1.8.0_66\bin\javaw.exe (Jun 23, 2016, 11:03:17 AM)
Hello world
```

5：RPC性能之王

Rpc	并发客户端	每客户端调用次数	总调用次数	执行时间	每秒调用数tps
ice	**1**	**300000**	**300000**	**16s**	**18329**
dubbo	1	300000	300000	52s	5675
thrift	1	300000	300000	23s	12832
grpc	1	300000	300000	77s	3896

Rpc	并发客户端	每客户端调用次数	总调用次数	执行时间	每秒调用数tps
ice	**100**	**300000**	**30000000**	**361s**	**83014**
dubbo	100	300000	30000000	1599s	18760
thrift	100	300000	30000000	597s	50211
grpc	100	300000	30000000	2186s	13721

从数据可以看出ice，thrift的tps最高，ice是thrift的1.6倍，是dubbo的4.4倍，是grpc的6倍，来自南哥测试报告,http://i.mycat.io

6 ：ICE RPC总结

很简洁，只依赖一个包

高性能，很稳定

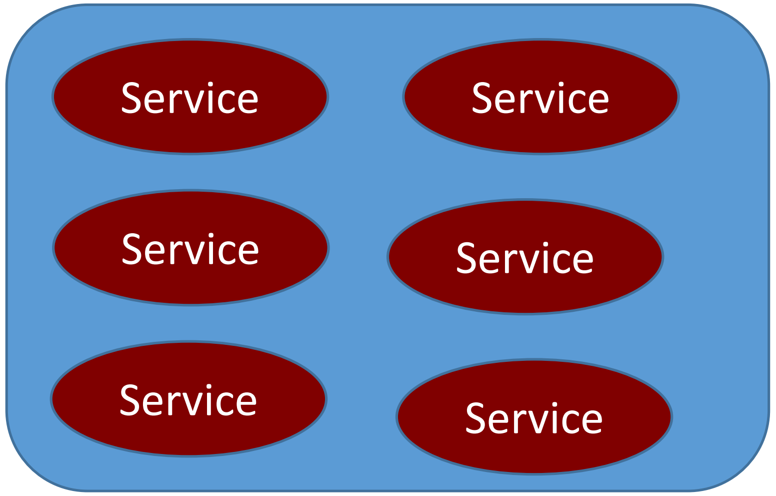
多语言支持

二：ICE微服务架构实践

1 : IceBox(1)

IceBox

```
java IceBox.Server --Ice.Config=config.icebox
```



=

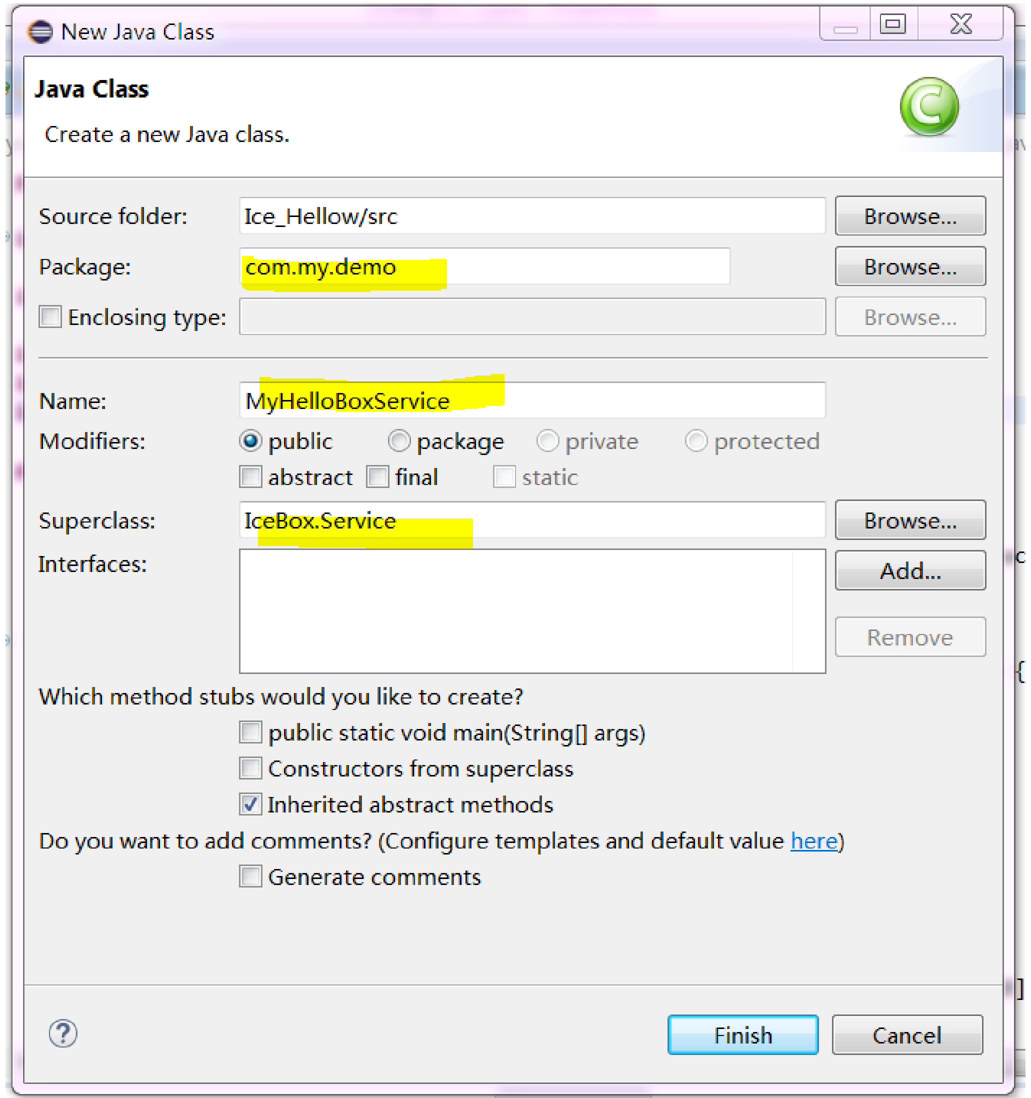
Apache Tomcat

```
public abstract interface IceBox.Service {
  public abstract void start(java.lang.String arg0, Ice.Communicator arg1, String[] arg2);
  public abstract void stop();
}
```

= Servlet

1：IceBox(2)

开发运行在IceBox中的Service



New Java Class

Java Class

Create a new Java class.

Source folder: Ice_Hellow/src

Package: com.my.demo

Enclosing type:

Name: MyHelloBoxService

Modifiers: [✓ public] package private protected

abstract final static

Superclass: IceBox.Service

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)

Constructors from superclass

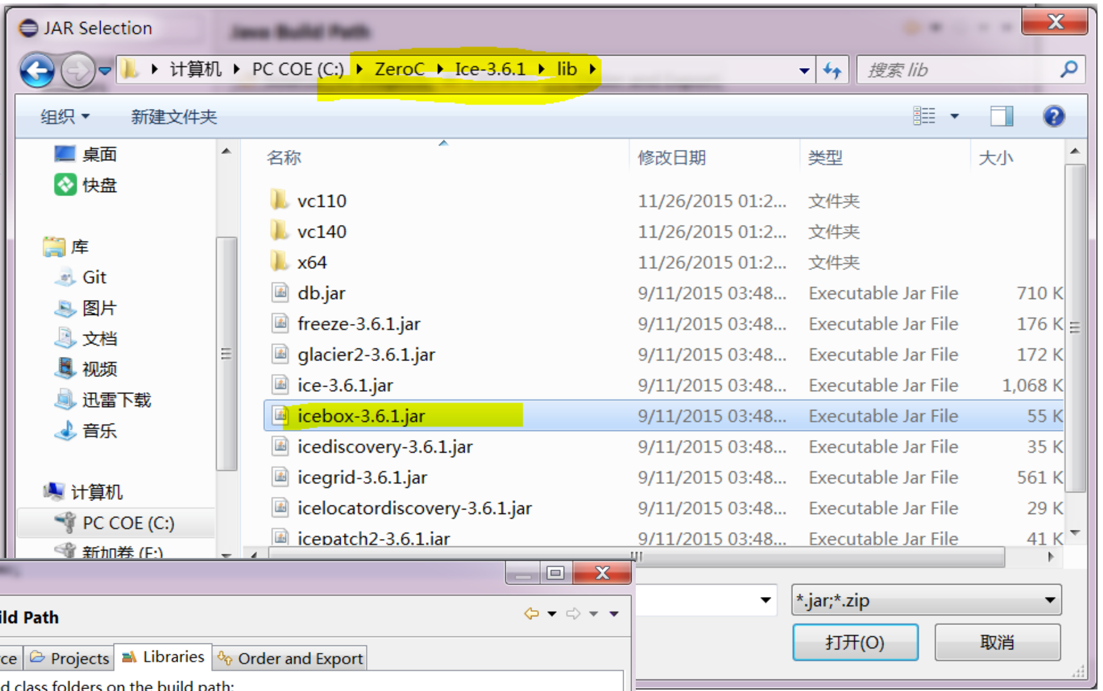
[✓ Inherited abstract methods]

Do you want to add comments? (Configure templates and default value here)

Generate comments

Browse... Browse... Browse... Browse... Add... Remove Finish Cancel

依赖包



JAR Selection

计算机 ▸ PC COE (C:) ▸ ZeroC ▸ Ice-3.6.1 ▸ lib ▸

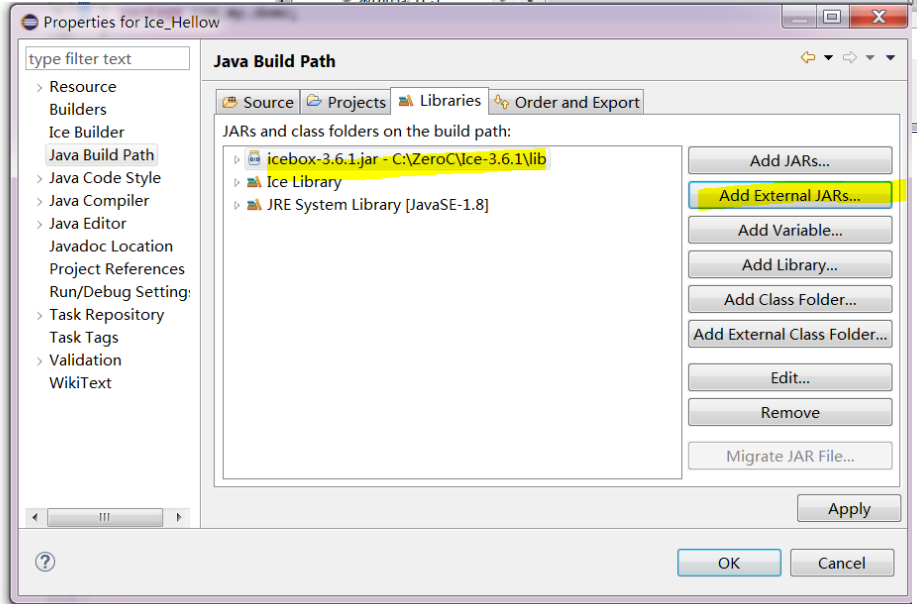
组织 ▾ 新建文件夹

桌面, 快盘, 库, Git, 图片, 文档, 视频, 迅雷下载, 音乐, 计算机, PC COE (C:), 新加卷 (E:)

名称	修改日期	类型	大小
vc110	11/26/2015 01:2...	文件夹	
vc140	11/26/2015 01:2...	文件夹	
x64	11/26/2015 01:2...	文件夹	
db.jar	9/11/2015 03:48...	Executable Jar File	710 K
freeze-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	176 K
glacier2-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	172 K
ice-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	1,068 K
icebox-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	55 K
icediscovery-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	35 K
icegrid-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	561 K
icelocatordiscovery-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	29 K
icepatch2-3.6.1.jar	9/11/2015 03:48...	Executable Jar File	41 K

.jar;.zip

打开(O) 取消



Properties for Ice_Hellow

type filter text

Resource, Builders, Ice Builder, Java Build Path, Java Code Style, Java Compiler, Java Editor, Javadoc Location, Project References, Run/Debug Setting, Task Repository, Task Tags, Validation, WikiText

Java Build Path

Source | Projects | Libraries | Order and Export

JARs and class folders on the build path:

- icebox-3.6.1.jar - C:\ZeroC\Ice-3.6.1\lib
- Ice Library
- JRE System Library [JavaSE-1.8]

Add JARs... | Add External JARs... | Add Variable... | Add Library... | Add Class Folder... | Add External Class Folder... | Edit... | Remove | Migrate JAR File...

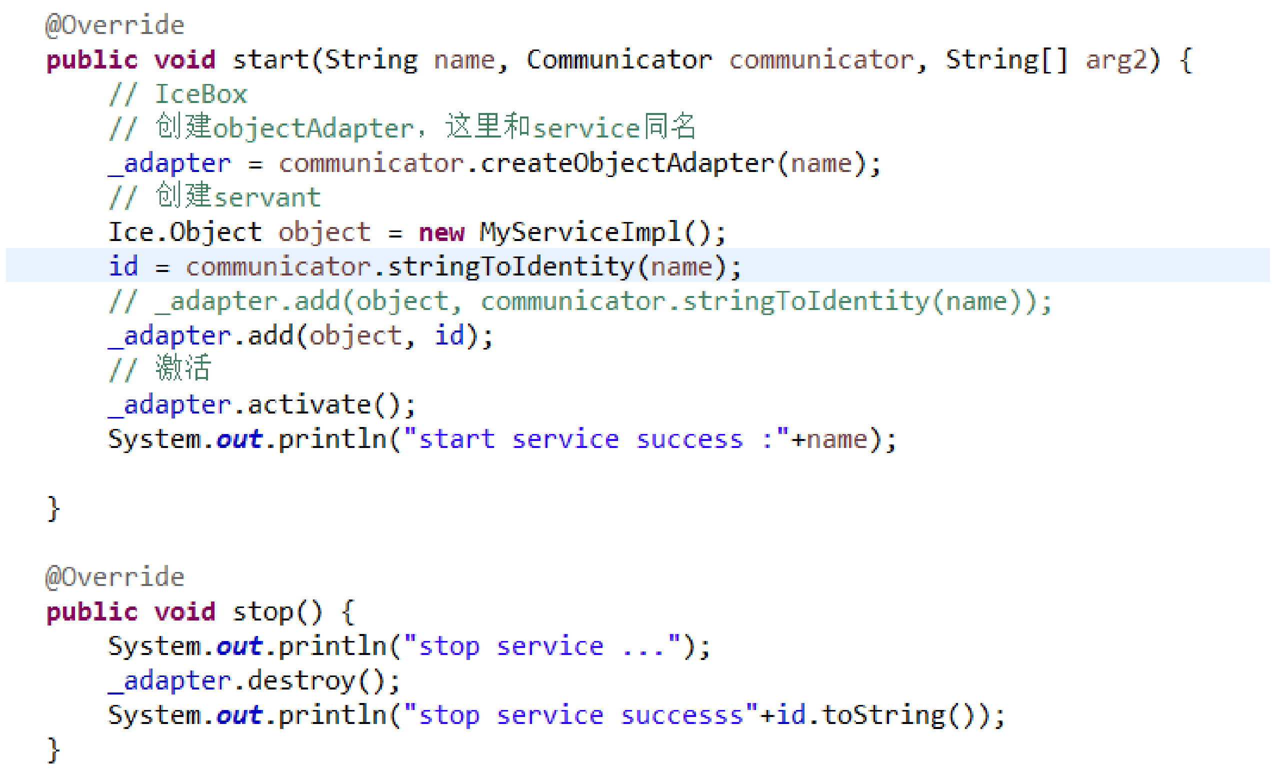
Apply

OK Cancel

1：IceBox(3)

开发运行在IceBox中的Service

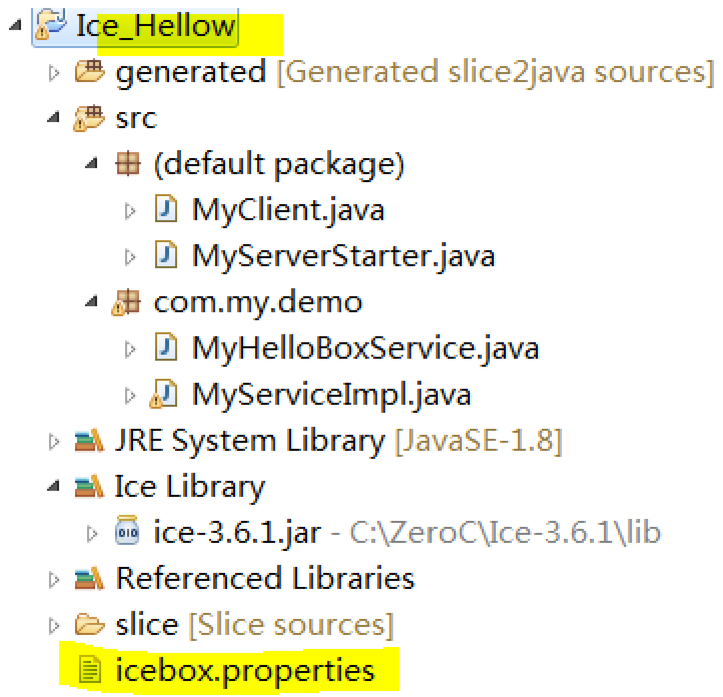
```java
package com.my.demo;
import Ice.Communicator;
import IceBox.Service;
public class MyHelloBoxService implements Service {
@Override
public void start(String arg0, Communicator arg1, String[] arg2) {
// TODO Auto-generated method stub

}
@Override
public void stop() {
// TODO Auto-generated method stub
}
}
```



```java
@Override
public void start(String name, Communicator communicator, String[] arg2) {
    // IceBox
    // 创建objectAdapter，这里和service同名
    _adapter = communicator.createObjectAdapter(name);
    // 创建servant
    Ice.Object object = new MyServiceImpl();
    id = communicator.stringToIdentity(name);
    // _adapter.add(object, communicator.stringToIdentity(name));
    _adapter.add(object, id);
    // 激活
    _adapter.activate();
    System.out.println("start service success :"+name);

}

@Override
public void stop() {
    System.out.println("stop service ...");
    _adapter.destroy();
    System.out.println("stop service successs"+id.toString());
}
```

1：IceBox(4)

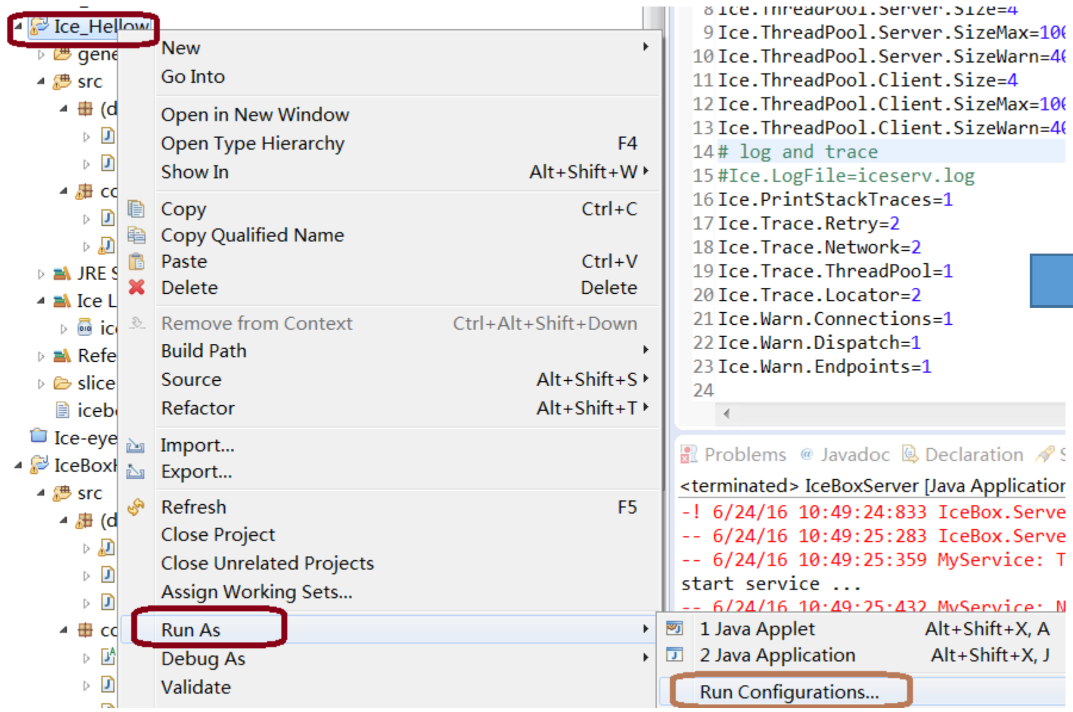
编写IceBox的配置文件



```
#server properties
IceBox.InheritProperties=1
IceBox.PrintServicesReady= MyAppIceBox 1
#service define begin
IceBox.Service.MyService=com.my.demo.MyHelloBoxService prop1=1 prop2=2 prop3=3
MyService.Endpoints=tcp -p 20000 -h localhost
#performance properties
Ice.ThreadPool.Server.Size=4
Ice.ThreadPool.Server.SizeMax=100
Ice.ThreadPool.Server.SizeWarn=40
Ice.ThreadPool.Client.Size=4
Ice.ThreadPool.Client.SizeMax=100
Ice.ThreadPool.Client.SizeWarn=40
#for system stronger
Ice.ACM.Client=300
Ice.ACM.Server=300
# log and trace
#Ice.LogFile=iceserv.log
Ice.PrintStackTraces=1
Ice.Trace.Retry=2
Ice.Trace.Network=2
Ice.Trace.ThreadPool=1
Ice.Trace.Locator=2
Ice.Warn.Connections=1
Ice.Warn.Dispatch=1
Ice.Warn.Endpoints=1
```

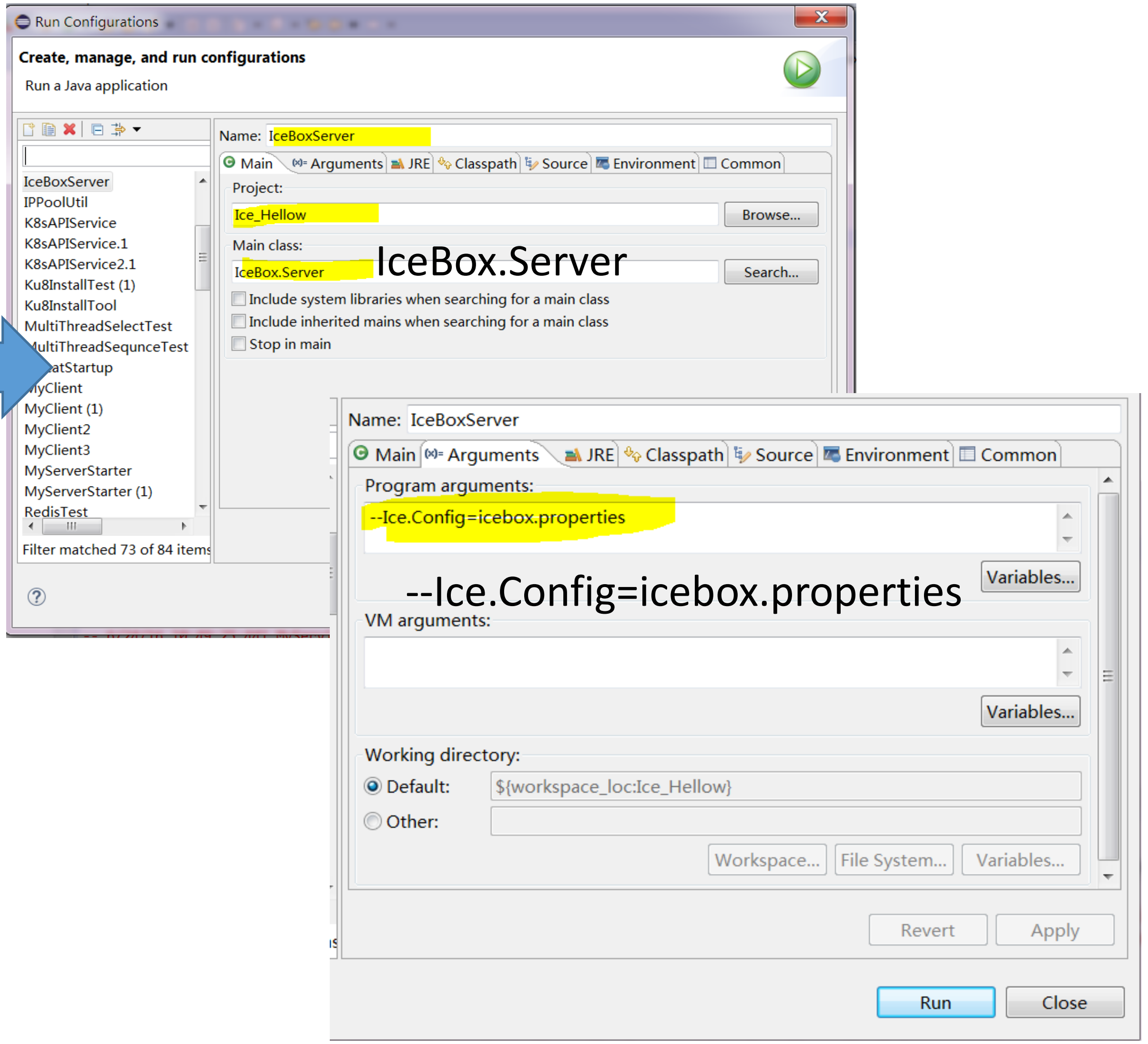
1：IceBox(5)

启动IceBox Server



```
Ice.ThreadPool.Server.Size=4
Ice.ThreadPool.Server.SizeMax=10
Ice.ThreadPool.Server.SizeWarn=4
Ice.ThreadPool.Client.Size=4
Ice.ThreadPool.Client.SizeMax=10
Ice.ThreadPool.Client.SizeWarn=4
# log and trace
#Ice.LogFile=iceserv.log
Ice.PrintStackTraces=1
Ice.Trace.Retry=2
Ice.Trace.Network=2
Ice.Trace.ThreadPool=1
Ice.Trace.Locator=2
Ice.Warn.Connections=1
Ice.Warn.Dispatch=1
Ice.Warn.Endpoints=1
```

```
<terminated> IceBoxServer [Java Applicatior
-! 6/24/16 10:49:24:833 IceBox.Serve
-- 6/24/16 10:49:25:283 IceBox.Serve
-- 6/24/16 10:49:25:359 MyService: T
start service ...
-- 6/24/16 10:49:25:432 MyService: N
```



Name: IceBoxServer

Project: Ice_Hellow

Main class: IceBox.Server

IceBox.Server

Include system libraries when searching for a main class

Include inherited mains when searching for a main class

Stop in main

Name: IceBoxServer

Program arguments: --Ice.Config=icebox.properties

--Ice.Config=icebox.properties

VM arguments:

Working directory:

[✓ Default:] ${workspace_loc:Ice_Hellow}

Other:

1：IceBox(6)

启动**IceBox Server**

```
-- 6/24/16 10:57:26:871 IceBox.Server: ThreadPool: creating Ice.ThreadPool.Client: Size = 4, SizeMax =
100, SizeWarn = 40
-- 6/24/16 10:57:26:950 MyService: ThreadPool: creating Ice.ThreadPool.Client: Size = 4, SizeMax = 100,
SizeWarn = 40
start service ...
-- 6/24/16 10:57:27:021 MyService: Network: attempting to bind to tcp socket 127.0.0.1:20000
-- 6/24/16 10:57:27:027 MyService: Network: listening for tcp connections
   local address = 127.0.0.1:20000
-- 6/24/16 10:57:27:032 MyService: ThreadPool: creating Ice.ThreadPool.Server: Size = 4, SizeMax = 100,
SizeWarn = 40
-- 6/24/16 10:57:27:035 MyService: Network: published endpoints for object adapter `MyService':
   tcp -h localhost -p 20000 -t 60000
-- 6/24/16 10:57:27:044 MyService: Network: accepting tcp connections at 127.0.0.1:20000
start service success
MyAppIceBox 1 ready
```

1：IceBox(7)

客户端发起调用

```
-- 6/24/16 10:58:26:910 IceBox.Server: ThreadPool: shrinking Ice.ThreadPool.Client: Size=3
-- 6/24/16 10:58:26:912 IceBox.Server: ThreadPool: shrinking Ice.ThreadPool.Client: Size=2
-- 6/24/16 10:58:26:914 IceBox.Server: ThreadPool: shrinking Ice.ThreadPool.Client: Size=1
-- 6/24/16 10:58:26:956 MyService: ThreadPool: shrinking Ice.ThreadPool.Client: Size=3
-- 6/24/16 10:58:26:957 MyService: ThreadPool: shrinking Ice.ThreadPool.Client: Size=2
-- 6/24/16 10:59:27:045 MyService: ThreadPool: shrinking Ice.ThreadPool.Server: Size=3
-- 6/24/16 10:59:27:048 MyService: ThreadPool: shrinking Ice.ThreadPool.Server: Size=2
-- 6/24/16 10:59:27:051 MyService: ThreadPool: shrinking Ice.ThreadPool.Server: Size=1
-- 6/24/16 11:01:26:964 MyService: ThreadPool: shrinking Ice.ThreadPool.Client: Size=1
-- 6/24/16 11:02:08:179 MyService: Network: trying to accept tcp connection
   local address = 127.0.0.1:20000
   remote address = 127.0.0.1:53700
-- 6/24/16 11:02:08:208 MyService: Network: accepted tcp connection
   local address = 127.0.0.1:20000
   remote address = 127.0.0.1:53700
-- 6/24/16 11:02:08:215 MyService: ThreadPool: growing Ice.ThreadPool.Server: Size=2
-- 6/24/16 11:02:08:233 MyService: Network: closed tcp connection
   local address = 127.0.0.1:20000
   remote address = 127.0.0.1:53700
```

1：IceBox(8)

管理IceBox Server

一：**icebox.properties**中增加下面的管理相关的配置

```
#定义IceBoxAdmin名称，默认是IceBox
Ice.Admin.InstanceName=Box
#Ice.Admin访问ServiceManager的端口
Ice.Admin.Endpoints=tcp -p 9998 -h 127.0.0.1
```

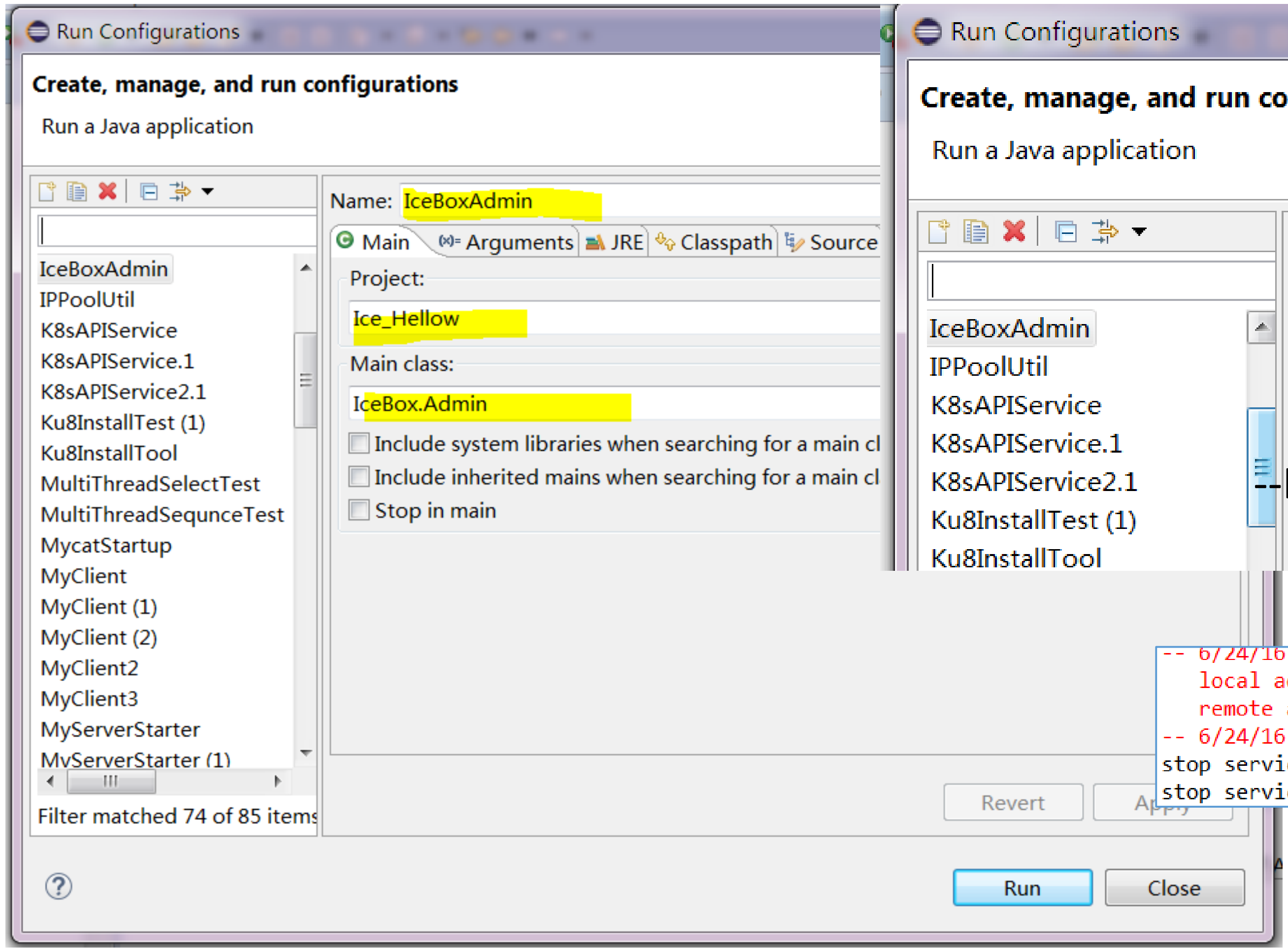
config.admin
icebox.properties

二：新建配置文件：**config.admin**

```
#这里的Box要和Ice.Admin.InstanceName对应。后面的端口要和Ice.Admin.Endpoints的配置相
IceBoxAdmin.ServiceManager.Proxy=Box/admin -f IceBox.ServiceManager:tcp -p 9998 -h 127.0.0.1
```

1：IceBox(9)

启动IceBox Admin进程



Run Configurations

Create, manage, and run configurations

Run a Java application

IceBoxAdmin
IPPoolUtil
K8sAPIService
K8sAPIService.1
K8sAPIService2.1
Ku8InstallTest (1)
Ku8InstallTool
MultiThreadSelectTest
MultiThreadSequnceTest
MycatStartup
MyClient
MyClient (1)
MyClient (2)
MyClient2
MyClient3
MyServerStarter
MyServerStarter (1)

Filter matched 74 of 85 items

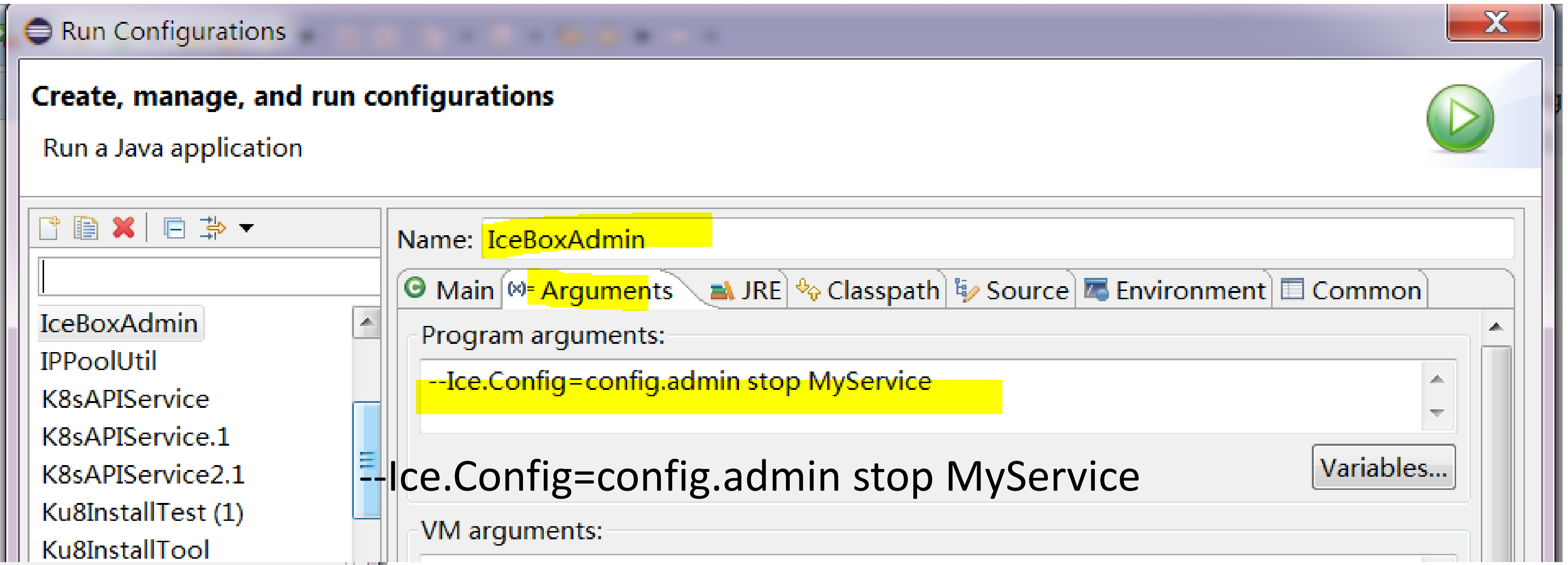
Name: IceBoxAdmin

Main | Arguments | JRE | Classpath | Source

Project: Ice_Hellow

Main class: IceBox.Admin

- [ ] Include system libraries when searching for a main cl
- [ ] Include inherited mains when searching for a main cl
- [ ] Stop in main



Run Configurations

Create, manage, and run configurations

Run a Java application

IceBoxAdmin
IPPoolUtil
K8sAPIService
K8sAPIService.1
K8sAPIService2.1
Ku8InstallTest (1)
Ku8InstallTool

Name: IceBoxAdmin

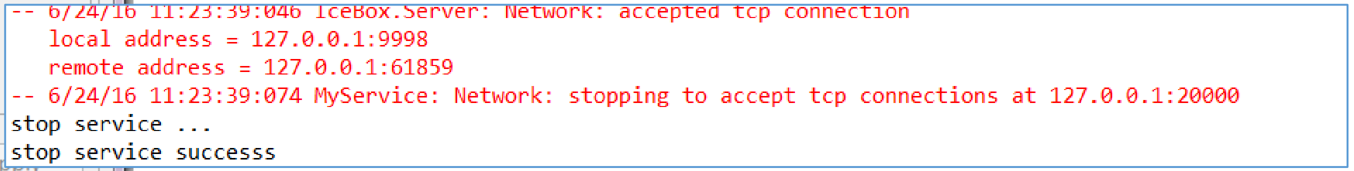
Main | Arguments | JRE | Classpath | Source | Environment | Common

Program arguments:

--Ice.Config=config.admin stop MyService

VM arguments:

--Ice.Config=config.admin stop MyService



```
-- 6/24/16 11:23:39:046 IceBox.Server: Network: accepted tcp connection
   local address = 127.0.0.1:9998
   remote address = 127.0.0.1:61859
-- 6/24/16 11:23:39:074 MyService: Network: stopping to accept tcp connections at 127.0.0.1:20000
stop service ...
stop service successs
```

1：IceBox 总结

是一个单独JVM进程

是一个可托管多个“服务”的Server

有API接口可远程管理

docker

下一期预告

深入IceGrid微服务架构实践

谢谢观看

Leader us高端Java培训报名群：434568702