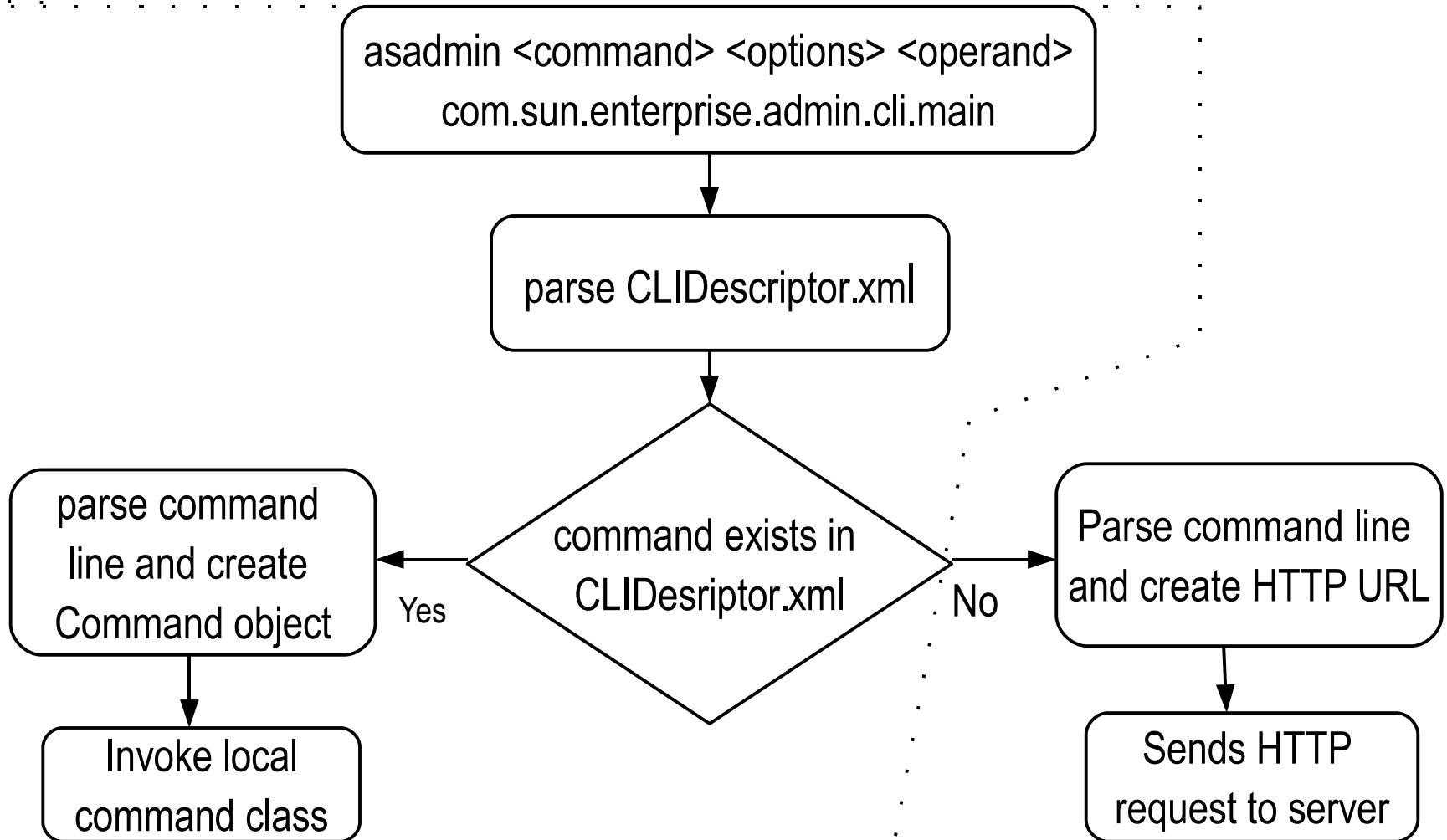# GlassFish V3
Admin CLI

Jane Young
janey@dev.java.net
January 14th 2008

Agenda

- Admin CLI
- Local Commands
- Remote Commands

Admin CLI

- v3/admin/cli and v3/admin/cli-framework modules

asadmin \<command\> \<options\> \<operand\>
com.sun.enterprise.admin.cli.main

parse CLIDescriptor.xml

command exists in CLIDesriptor.xml

parse command line and create Command object

Yes

Invoke local command class

No

Parse command line and create HTTP URL

Sends HTTP request to server

Local Commands

- Same as in GF V2

Remote Commands

- HTTP URL Get/Post – REST Web Services
- For Admin CLI Commands, the HTTP URL is:

  http://<host>:<port>/__asadmin/<command>?<option-name>=<option-value>?DEFAULT=<operand>

- URI containing "__asadmin" invokes the AdminAdapter class on the server side (Grizzly Adapter)
- AdminAdapter class searches for the command class in the inhabitat
- executes the command class

Remote Command Class

- Located in the module repository

- Implements AdminCommand class

- @Service annotation – name of the command -- Gfv3 depends on this information to find the Service.

- @Param annotation – command option

- @Param (primary=true) – command operand

- @Inject – inject dependent Service (i.e. @Inject

    Domain domain;)

Remote Command Class – cont'd

- Example

```
package com.foo.bar;
@Service(name="dummycommand")
@Param
String option1;
@Param(optional=true, shortName="o")
String option2;
@Param(primary=true)
String operand;
public class DummyCommand implements AdminCommand {
  public void execute(AdminCommandContext context) {
    ...
  }
}
```

7

Remote Command Class – cont'd

- AdminCommandContext
  - > Utility containing useful services
    - >ActionReport
    - >CommandParameters
    - >Logger

Remote Command Class – cont'd

- ## What is returned to the client?
  - > From CLI: java.util.jar.Manifest
    - > where message is saved in the Manifest's main Attribute: getMainAttribute().putValue("message", <value>);
  - > From browser: HTML

Remote Command -- Why?

- Location independent – loadable from any module

- Extensible – no preset list of commands, commands are discovered on demand

- Lightweight client – no need to add command jar files in the classpath of asadmin

- Injection dependency – use injection to express dependencies

- CLI via browser