java.nio.file
for remote systems

Vincent Partington - XebiaLabs

Jeroen van Erp - XebiaLabs

XebiaLabs

the deployment automation company

Jeroen van Erp
@hierynomus

Vincent
Partington
@vpartington

XebiaLabs
the deployment automation company

Our Company. Our Product.

XebiaLabs

- Headquartered in Boston, Massachusetts
- Global development and support offices in NL, UK, France and India

Deployit

Our flagship product, Deployit, is a market-leading Application Release Automation platform.

Benefits include:
- Reduce development applications costs
- Accelerate application time to market
- Bridge the gap between Development and Operations

XebiaLabs

Java New I/O File

Recap: java.nio.file aka NIO.2

More abstract

More modular

java.nio.file.Path

java.nio.file.Files

java.nio.file.spi.FileSystemProvider

XebiaLabs
the deployment automation company

java.nio.Path

Methods	
Modifier and Type	**Method and Description**
int	**compareTo**(`Path` other) Compares two abstract paths lexicographically.
boolean	**endsWith**(`Path` other) Tests if this path ends with the given path.
boolean	**endsWith**(`String` other) Tests if this path ends with a `Path`, constructed by converting path string, in exactly the manner specified by the **endsWith** method.
boolean	**equals**(`Object` other) Tests this path for equality with the given object.
`Path`	**getFileName**() Returns the name of the file or directory denoted by this path as a object.
`FileSystem`	**getFileSystem**() Returns the file system that created this object.
`Path`	**getName**(int index) Returns a name element of this path as a `Path` object.
int	**getNameCount**() Returns the number of name elements in the path.
`Path`	**getParent**() Returns the *parent path*, or `null` if this path does not have a parent.
`Path`	**getRoot**() Returns the root component of this path as a `Path` object, or `null` if this path does not have a root component.
int	**hashCode**()

Immutable!

NOT a File!

XebiaLabs
the deployment automation company

java.nio.Path

```java
FileSystem fileSystem = FileSystems.getDefault();
Path myHomeDir = fileSystem.getPath("/Users/ajvanerp");
Path notAFile = fileSystem.getPath("i", "do", "not", "exist");
```

java.nio.file.Files

Methods

Modifier and Type	Method and Description
static long	**copy**(**InputStream** in, **Path** target, **CopyOption**... options) Copies all bytes from an input stream to a file.
static long	**copy**(**Path** source, **OutputStream** out) Copies all bytes from a file to an output stream.
static **Path**	**copy**(**Path** source, **Path** target, **CopyOption**... options) Copy a file to a target file.
static **Path**	**createDirectories**(**Path** dir, **FileAttribute**<?>... attrs) Creates a directory by creating all nonexist parent directories first.
static **Path**	**createDirectory**(**Path** dir, **FileAttribute**<?>... attrs) Creates a new directory.
static **Path**	**createFile**(**Path** path, **FileAttribute**<?>... attrs) Creates a new and empty file, failing if the file already exists.
static **Path**	**createLink**(**Path** link, **Path** existing) Creates a new link (directory entry) for an existing file *(optional operation)*.
static **Path**	**createSymbolicLink**(**Path** link, **Path** target, **FileAttribute**<?

Use static imports!

java.nio.file.Files

```java
FileSystem fileSystem = FileSystems.getDefault();
Path hostsFile = fileSystem.getPath("/etc/hosts");
Path path = fileSystem.getPath("/Users/ajvanerp/hosts.txt");
copy(hostsFile, path, StandardCopyOption.REPLACE_EXISTING);
```

java.nio.file.spi.FileSystemProvider

Methods

Modifier and Type	Method and Description
abstract void	**checkAccess**(**Path** path, **AccessMode**... modes) Checks the existence, and optionally accessibility, of a file.
abstract void	**copy**(**Path** source, **Path** targ... **CopyOption**... options) Copy a file to a target file.
abstract void	**createDirectory**(**Path** dir, **FileAttribute**<?>... attrs) Creates a new directory.
void	**createLink**(**Path** link, **Path** exi... Creates a new link (directory entry) for an ... file.
void	**createSymbolicLink**(**Path** link, **Path** target, **FileAttribute**<? >... attrs) Creates a symbolic link to a target.
abstract void	**delete**(**Path** path) Deletes a file.
boolean	**deleteIfExists**(**Path** path) Deletes a file if it exists.
abstract <V extends **FileAttributeView**> V	**getFileAttributeView**(**Path** path, **Class**<V> type, **LinkOption**... options) Returns a file attribute view of a given type.
abstract **FileStore**	**getFileStore**(**Path** path) Returns the **FileStore** representing the file store where a file is located.

Service Provider Interface

Has a URI scheme like: file:///

XebiaLabs
the deployment automation company

Demo: Plain old NIO.2 + Process

- Copy GlassFish server zip to machine
- Unzip GlassFish zip
- Start GlassFish server
- Copy PetClinic ear to GlassFish
- Show log file
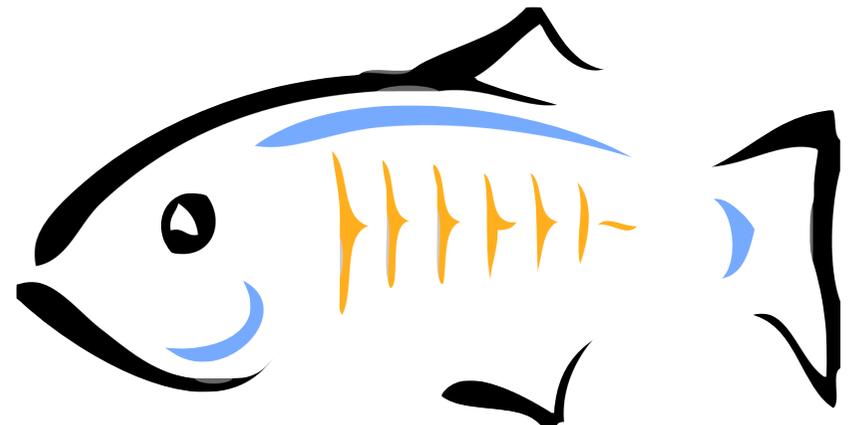
The problem
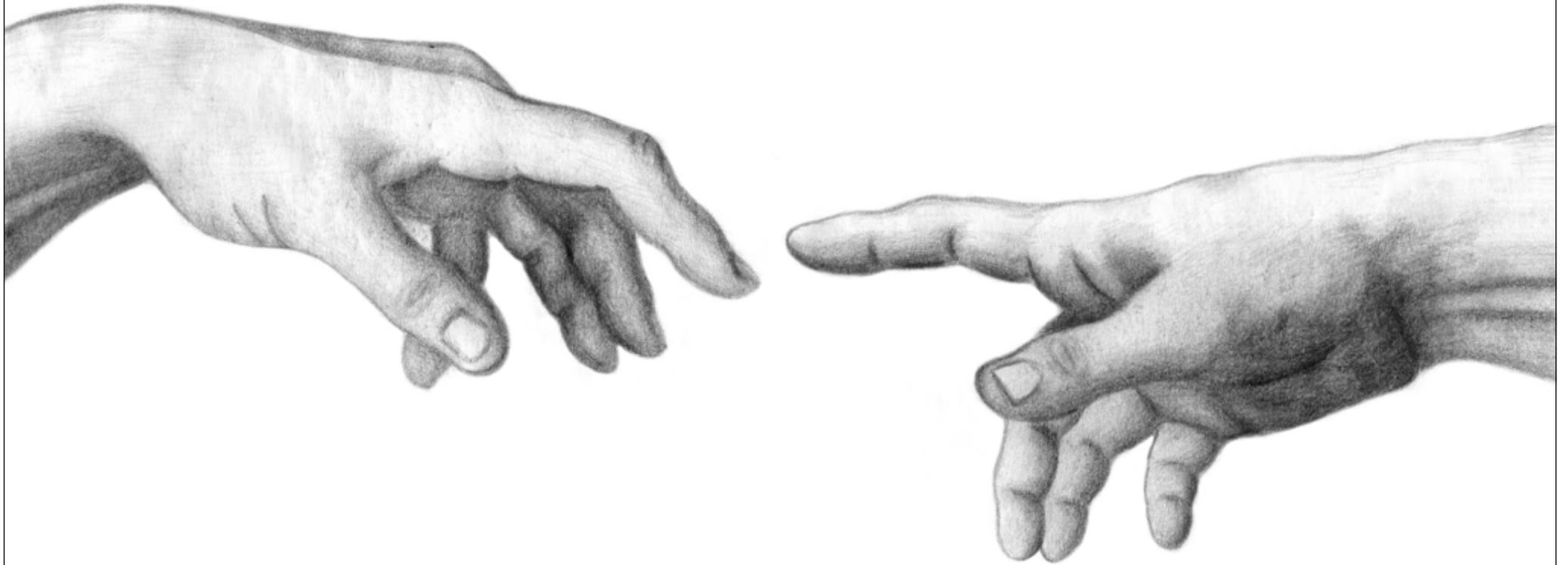
Overthere

The solution: Overthere

java.io.File   java.lang.Process

The solution: Overthere

OverthereFile ile

ja **OverthereProcess**

Interfaces

SPI

Remote Systems

nio-overthere

c.x.o.n.f.**OverthereFileSystemProvider**

c.x.o.n.f.**OvertherePath**

c.x.o.n.p.**Processes**

nio-overthere

Schema per protocol

Wraps Overthere

c.x.o.n.f.OverthereFileSystemPr
c.x.o.n.f.OvertherePath
c.x.n.o.p Processes

Adds NIO Process abstraction

XebiaLabs
the deployment automation company

local://

- Copy GlassFish server zip to machine
- Unzip GlassFish zip
- Start GlassFish server
- Copy PetClinic ear to GlassFish
- Show log file

SSH

- Secure
  - File transfer
  - Command execution
- Uses SSH/J
  - https://github.com/shikhar/sshj

SSH
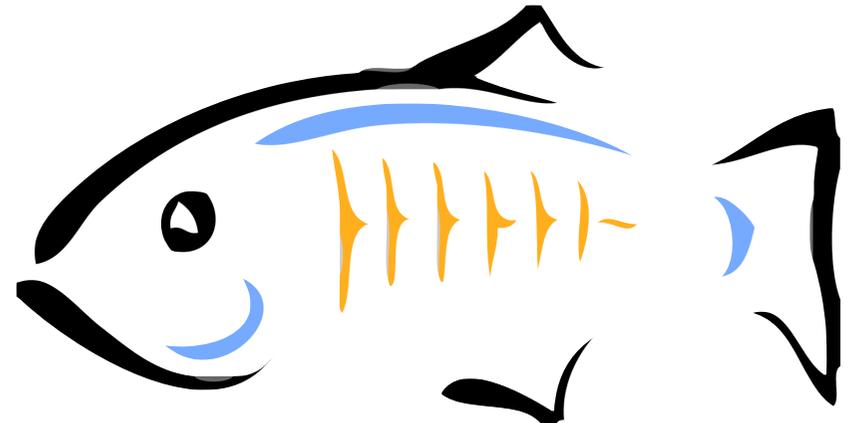
- Secure
  - File transfer
  - Command execution
- Uses SSH/J
  - https://github.com/shikhar/sshj

SCP vs. SFTP

Interactive auth
Password
Public key

Tunnels

ssh+sftp://

- Copy GlassFish server zip to machine
- Unzip GlassFish zip
- Start GlassFish server
- Copy PetClinic ear to GlassFish
- Show log file
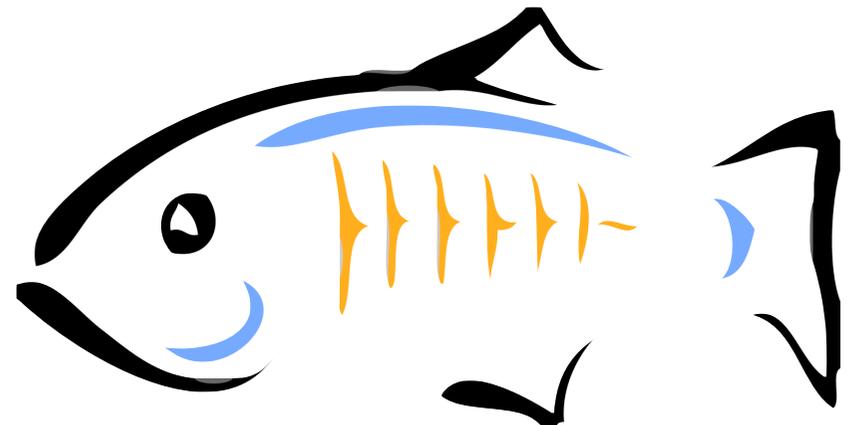
SSH + SUDO
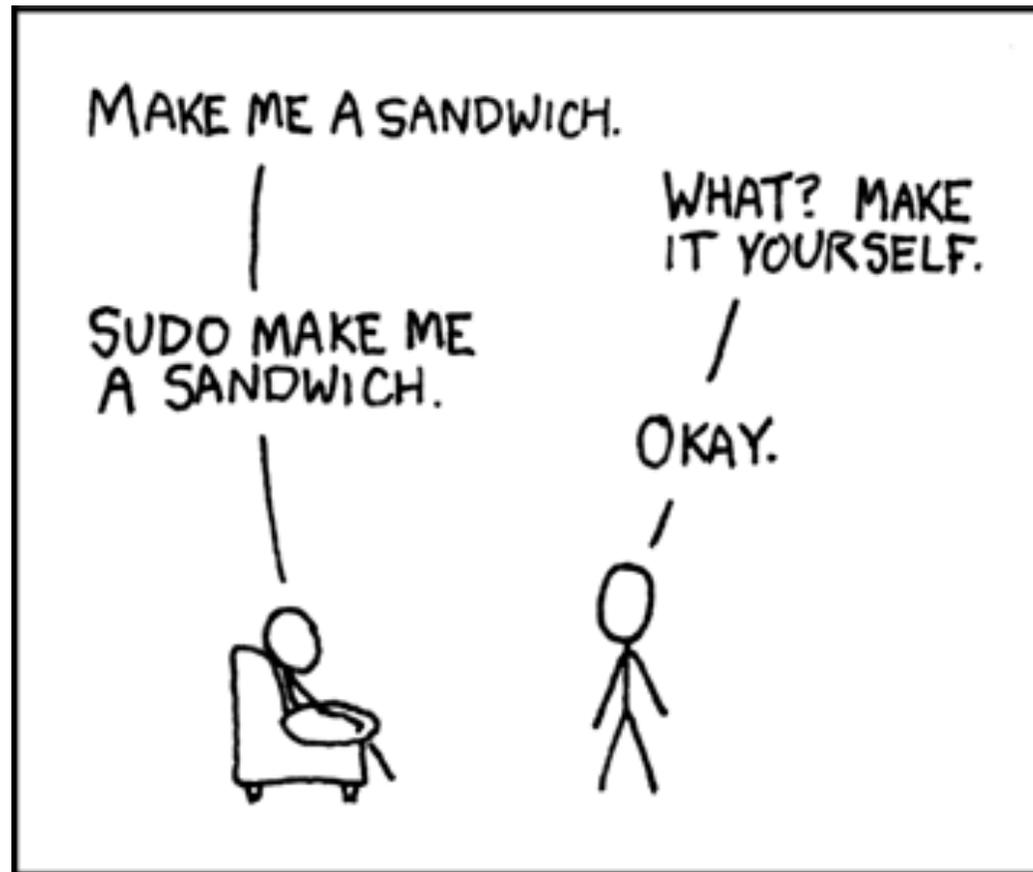
source: xkcd.com
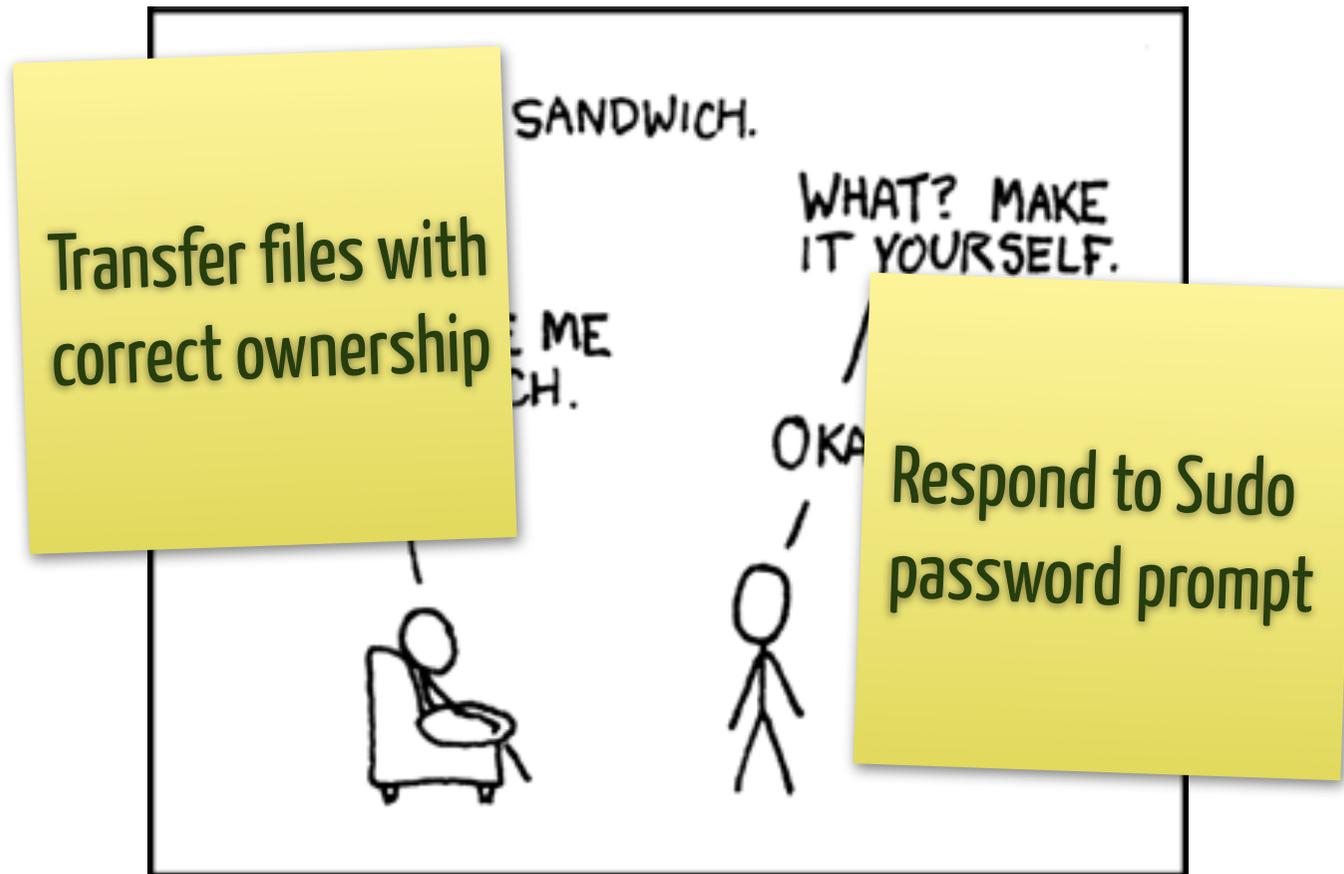
SSH + SUDO

source: xkcd.com

ssh+sudo://

- Copy GlassFish server zip to machine
- Unzip GlassFish zip
- Start GlassFish server
- Copy PetClinic ear to GlassFish
- Show log file
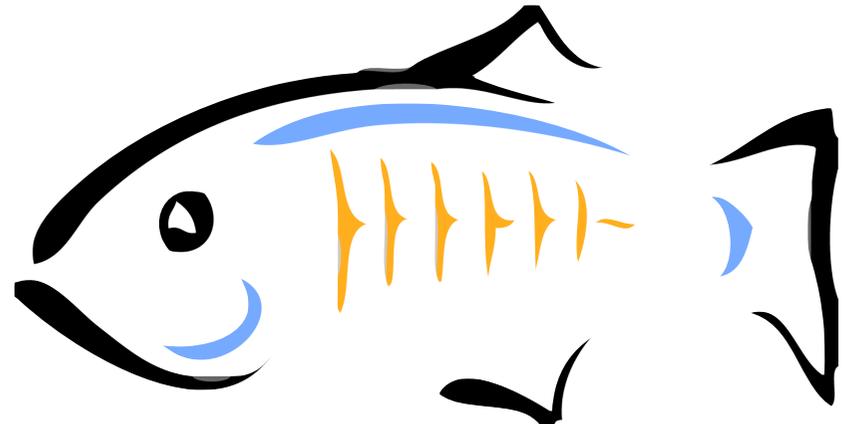
What about Windows?

SMB / CIFS

Telnet

WinRM

CIFS

- Authentication
  - Local accounts (basic auth)
  - Domain accounts (NTLM)
- Uses JCIFS library
  - http://jcifs.samba.org/

CIFS

- Authentication
  - Local accounts (basic auth)
  - Domain accounts (NTLM)
- Uses JCIFS library
  - http://jcifs.samba.org/

No Remote Execution

XebiaLabs
the deployment automation company

cifs+telnet://

- Copy GlassFish server zip to machine
- Unzip GlassFish zip
- Start GlassFish server
- Copy PetClinic ear to GlassFish
- Show log file
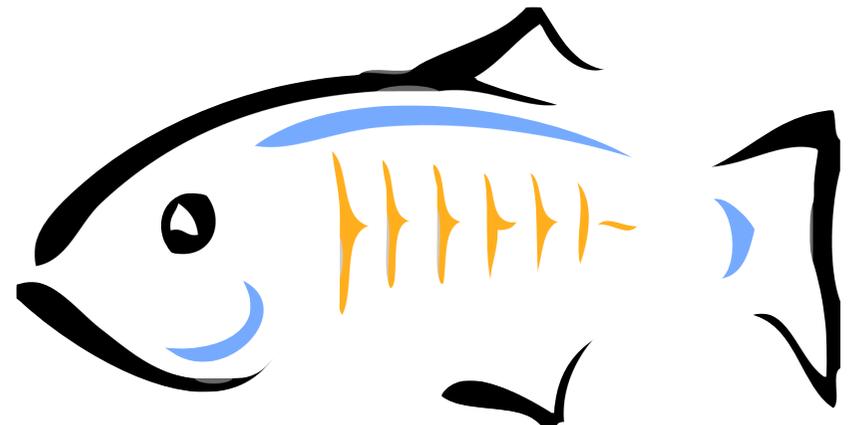
WinRM

- WS-Management (SOAP)
- HTTP and HTTPS
- Basic auth and Kerberos

WinRM

- WS-Management (SOAP)
- HTTP and HTTPS
- Basic auth and Kerberos

No XML Encryption (yet)

Pull request welcome ;-)

XebiaLabs
the deployment automation company

cifs+winrm://

- Copy GlassFish server zip to machine
- Unzip GlassFish zip
- Start GlassFish server
- Copy PetClinic ear to GlassFish
- Show log file
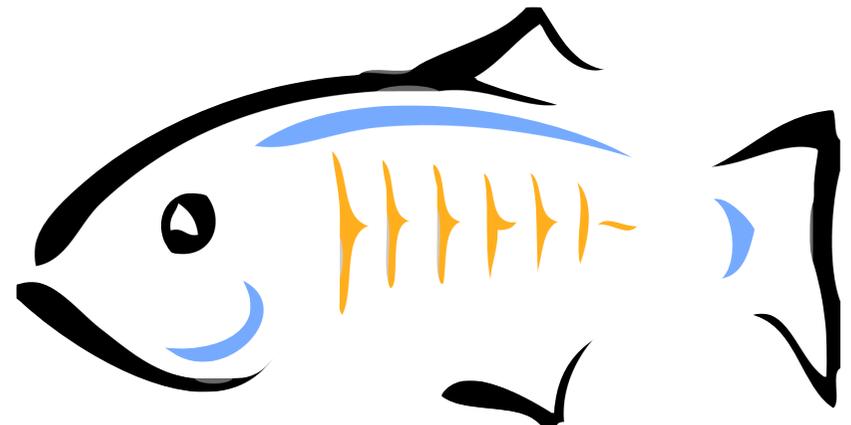
Conclusions

NIO experiences

Easy to use

NO process abstraction

URI schemes are nice

FileSystemProvider is the only stateful class

A lot of implementation work

XebiaLabs
the deployment automation company

Overthere's NIO interface allows you to write remote management code that is portable not only across **source** platforms, but also across **target** platforms

More information

Overthere:
https://github.com/xebialabs/overthere/

NIO-Overthere:
https://github.com/xebialabs/nio-overthere/

More information

Jeroen van Erp
@hierynomus
jvanerp@xebialabs.com

Vincent Partington
@vpartington
vpartington@xebialabs.com

XebiaLabs
the deployment automation company

Questions