

Jetty 9 Features - 2014

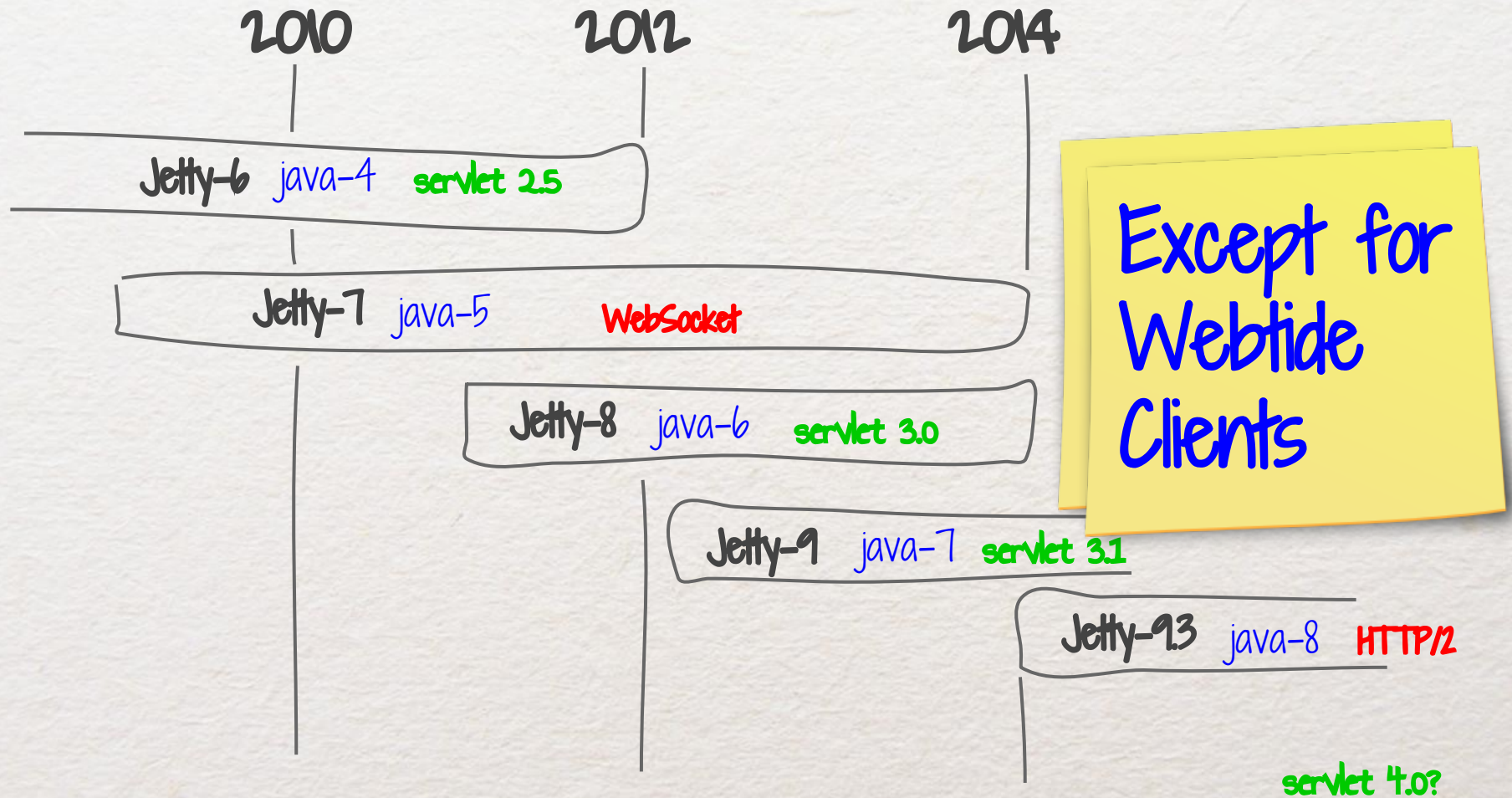
Greg Wilkins <gregw@webtide.com>

Founder: Jetty, cometd, Mortbay, Webtide

Standards: Servlet, Websocket, HTTP/2

<http://www.webtide.com>

Jetty Versions



Porting to Jetty-9

- Only ServerConnector
 - Connector uses ConnectionFactories
 - HTTP1
 - SSL, HTTP1
 - ALPN, SSL, HTTP1|SPDY|HTTP2
- Apache JSP
 - Recompile precompiled JSPs
- Distribution Modules / Base
- New FEATURES!

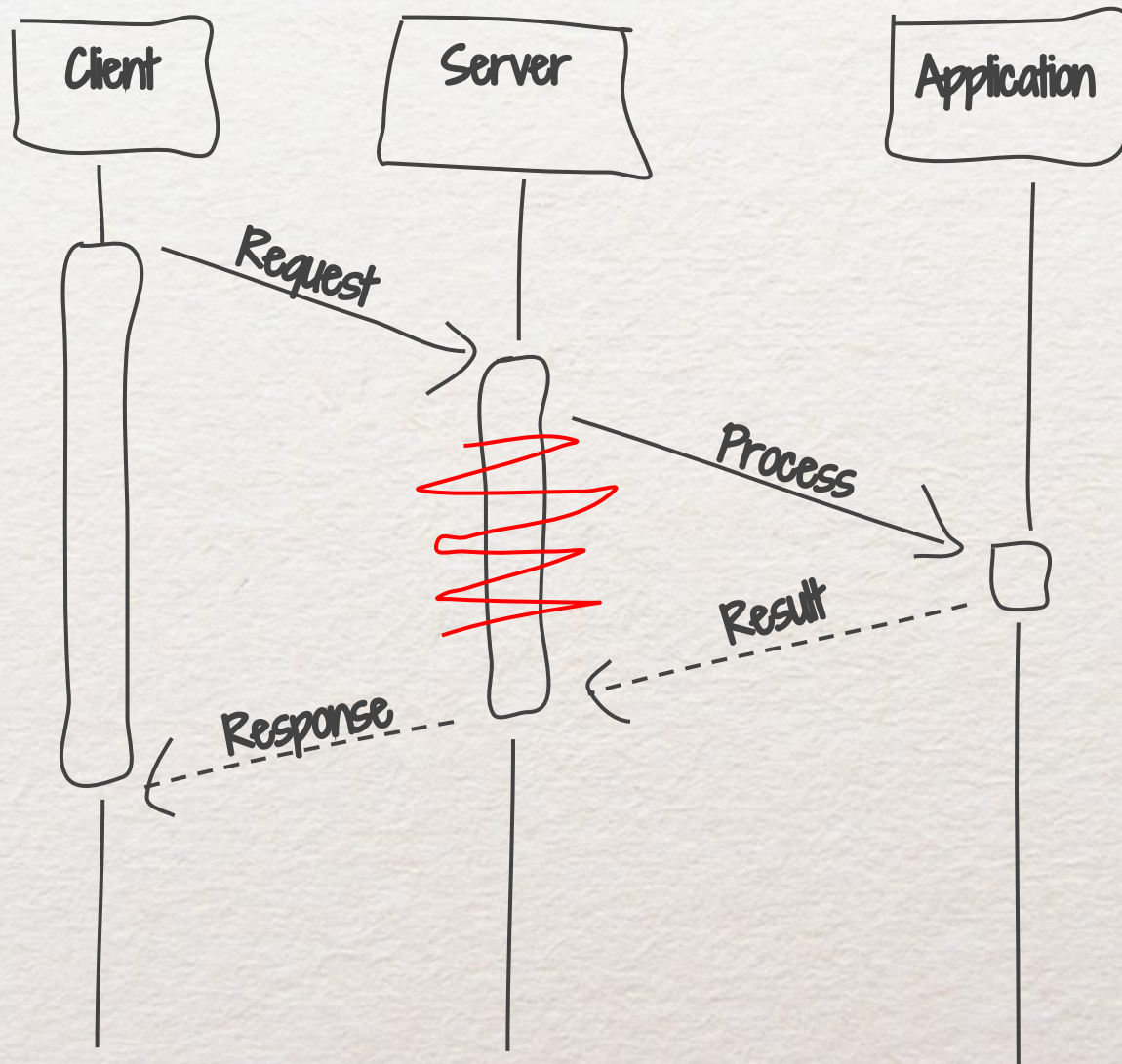
Apache JSP!

- More actively maintained
- Wider usage
- Contributed to Apache Project

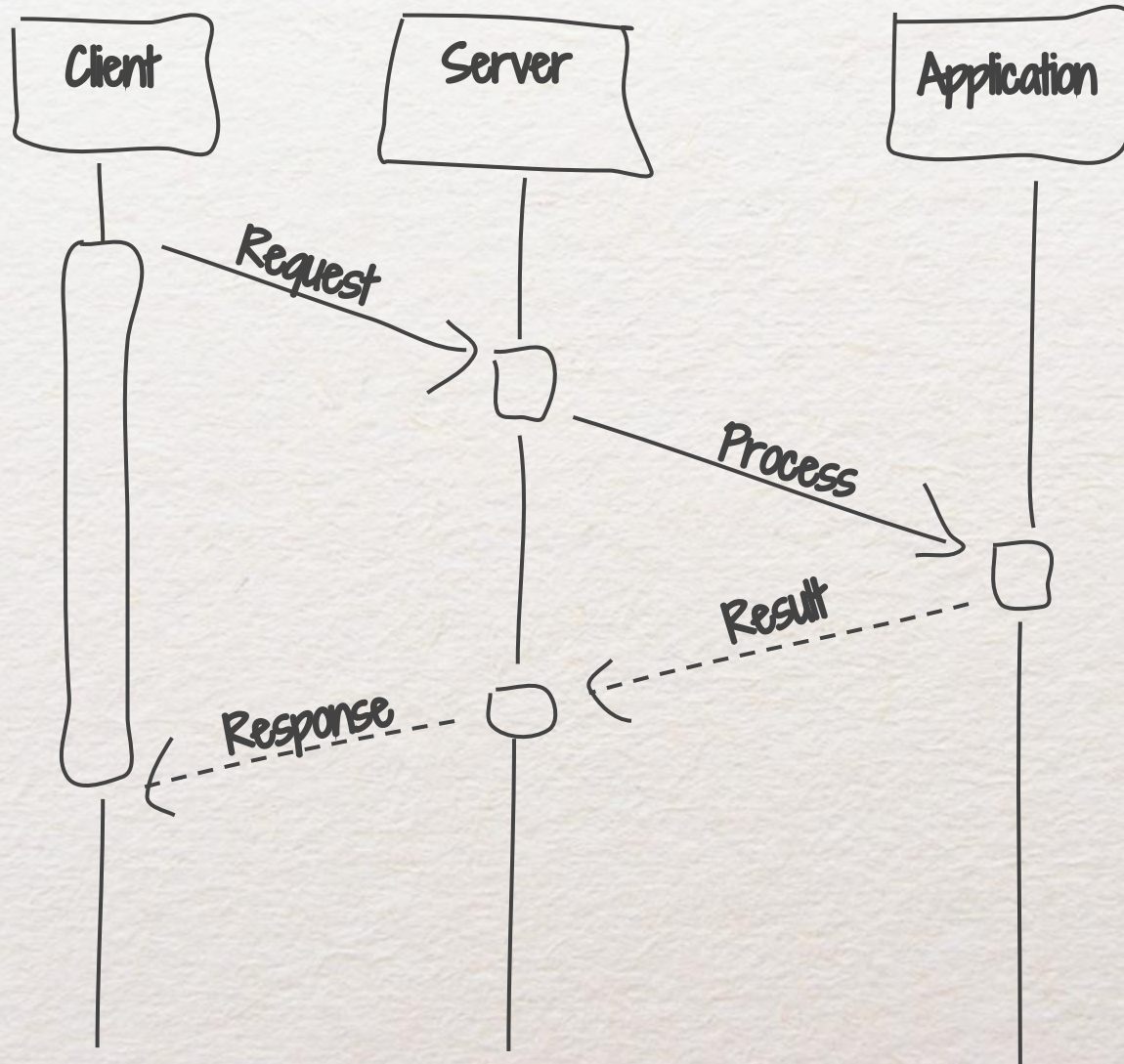
Async Servlet IO

Servlet 3.1

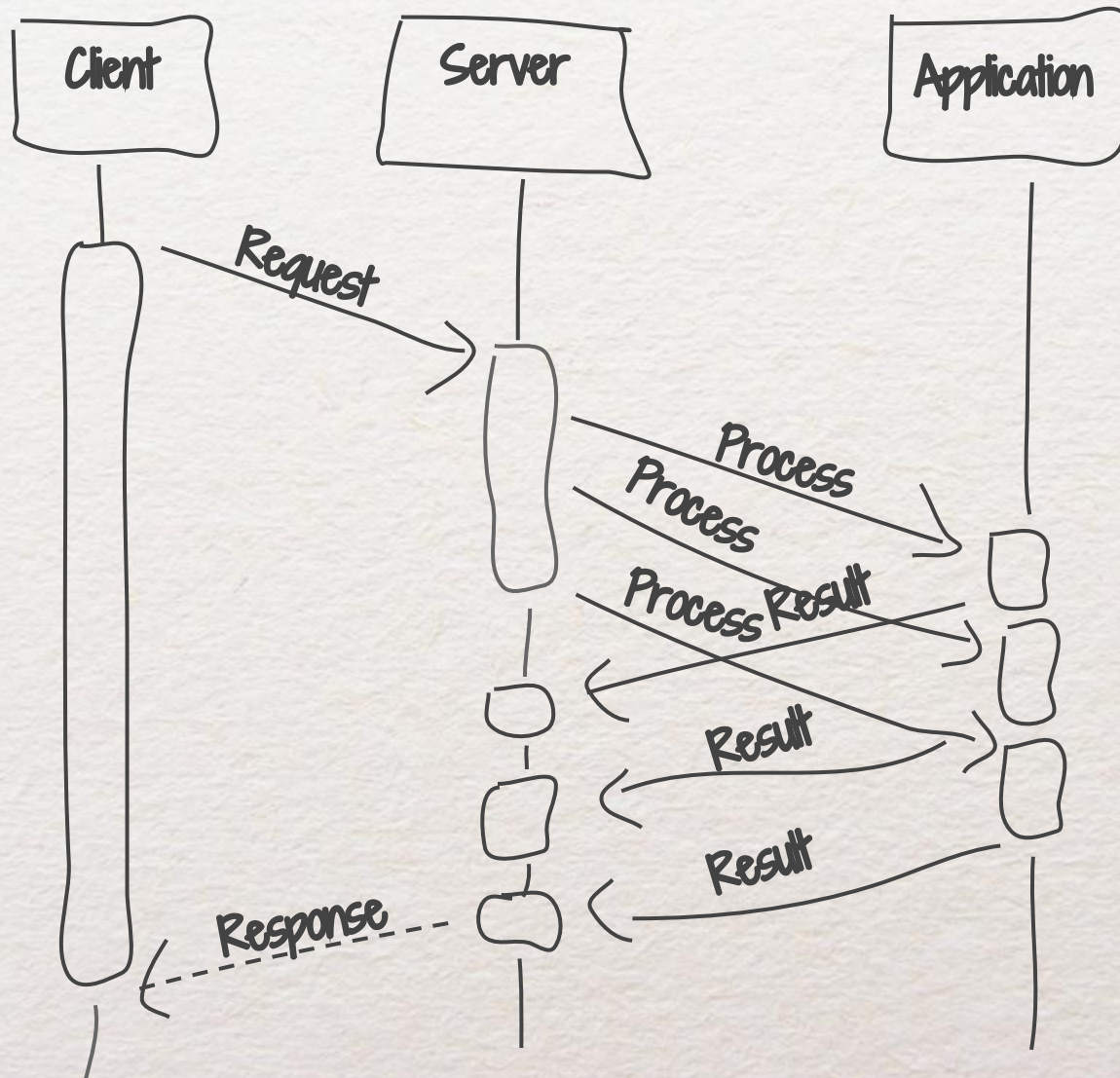
Synchronous Processing



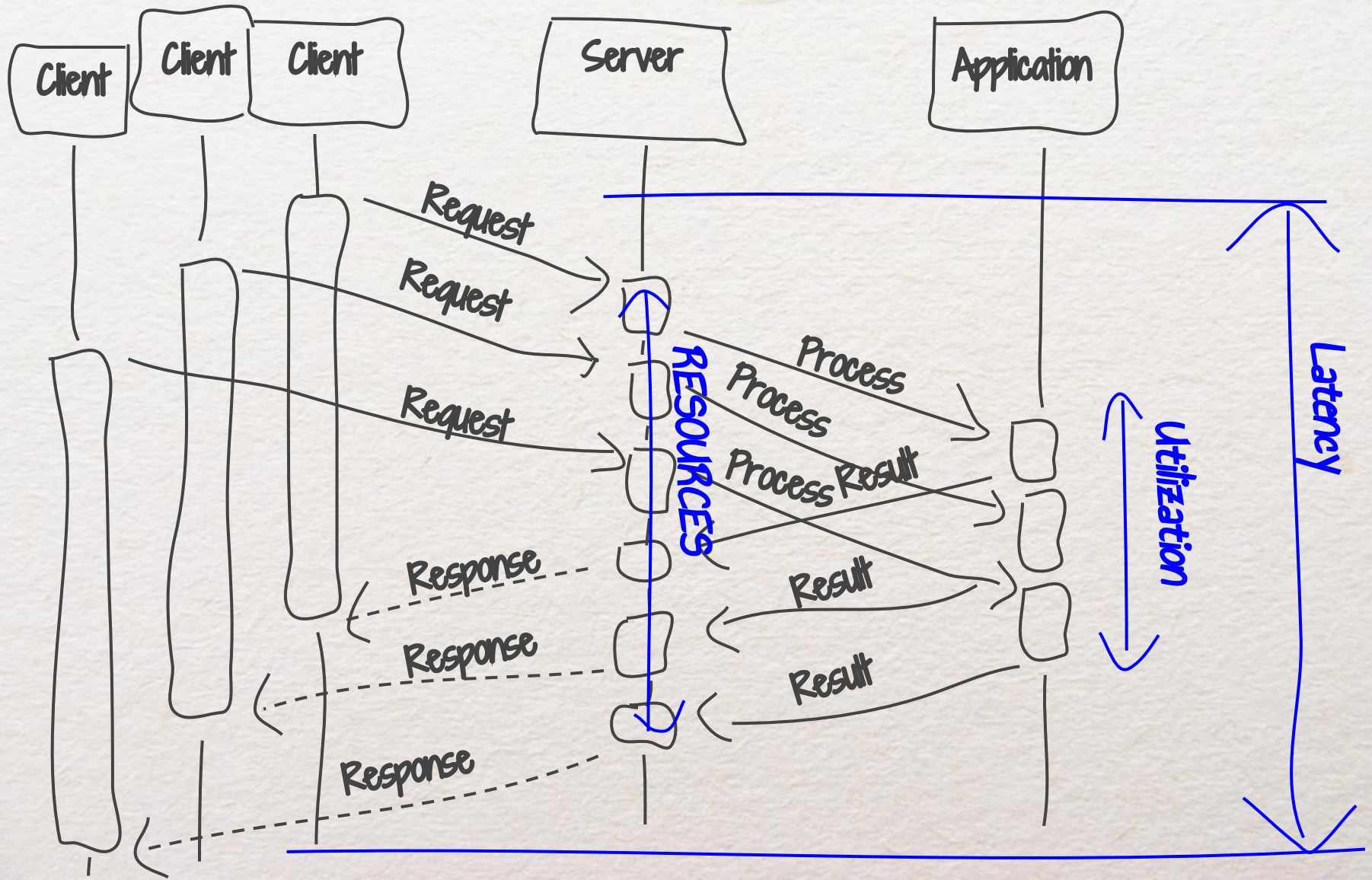
Asynchronous Processing



Asynchronous Processing



Asynchronous Processing



Servlet 3.1 Async Output API

```
class ServletOutputStream extends OutputStream
```

```
{
```

```
    void setWriteListener(WriteListener l);
```

```
    boolean isReady();
```

```
    ...
```

```
}
```

```
interface WriteListener extends EventListener
```

```
{
```

```
    void onWritePossible() throws IOException;
```

```
    void onError(Throwable t);
```

```
}
```


An Async File Servlet

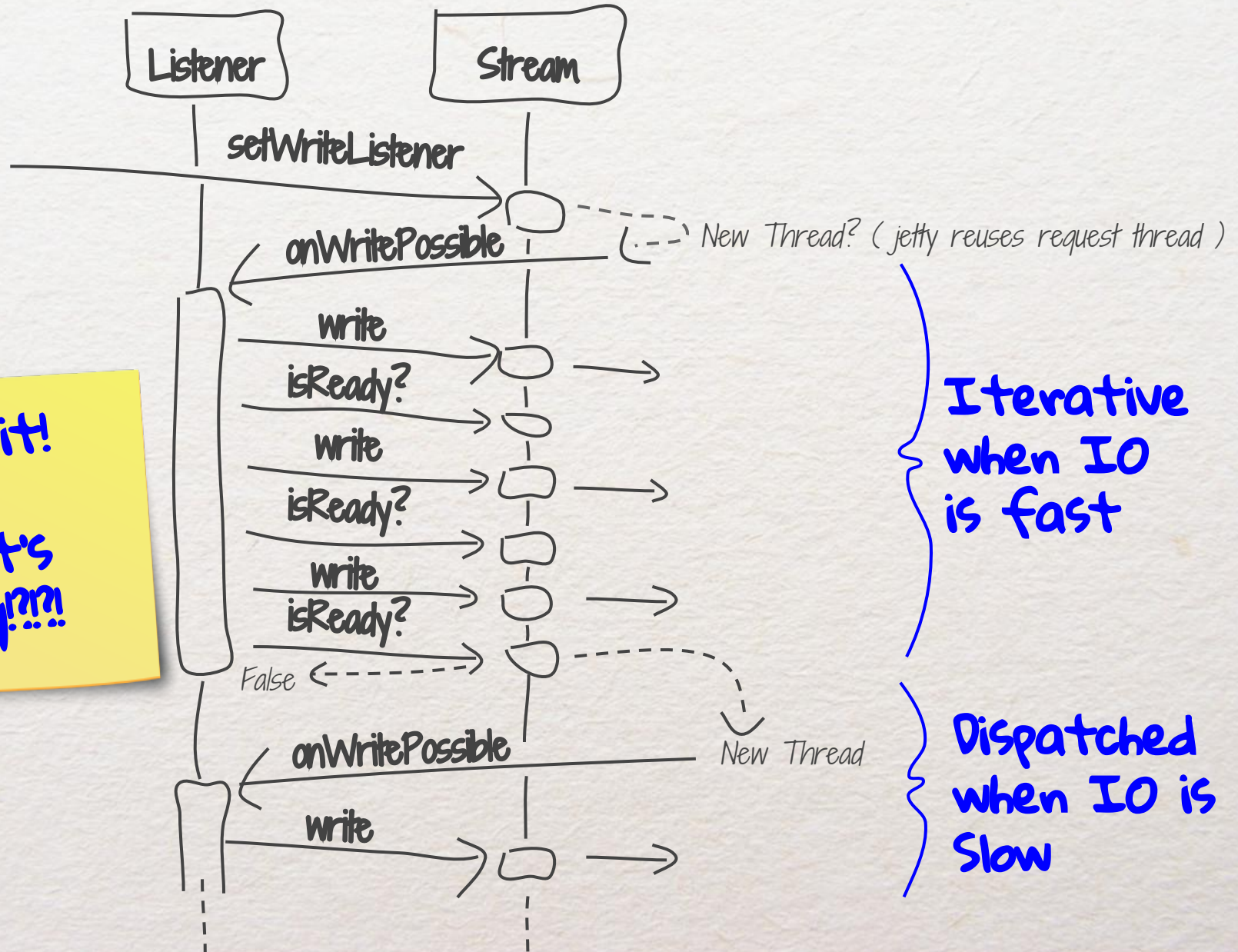
```
void doGet(HttpServletRequest req, final HttpServletResponse resp)
{
    final AsyncContext ctx=req.startAsync();
    final ServletOutputStream out = resp.getOutputStream();
    final FileInputStream file = ...;
    out.setWriteListener(new WriteListener()
    {
        void onWritePossible() throws IOException
        {
            while(out.isReady())
            {
                int len=file.read(buf);
                if (len<0)
                {
                    ctx.complete();
                    return;
                }
                out.write(buf, 0, len);
            }
        }
    });
}
```

active methods!
may callback onWritePossible

Iterative for fast IO!!

Threaded for slow IO!!

Iterative Servlet IO



Got it!

That's
easy...!!

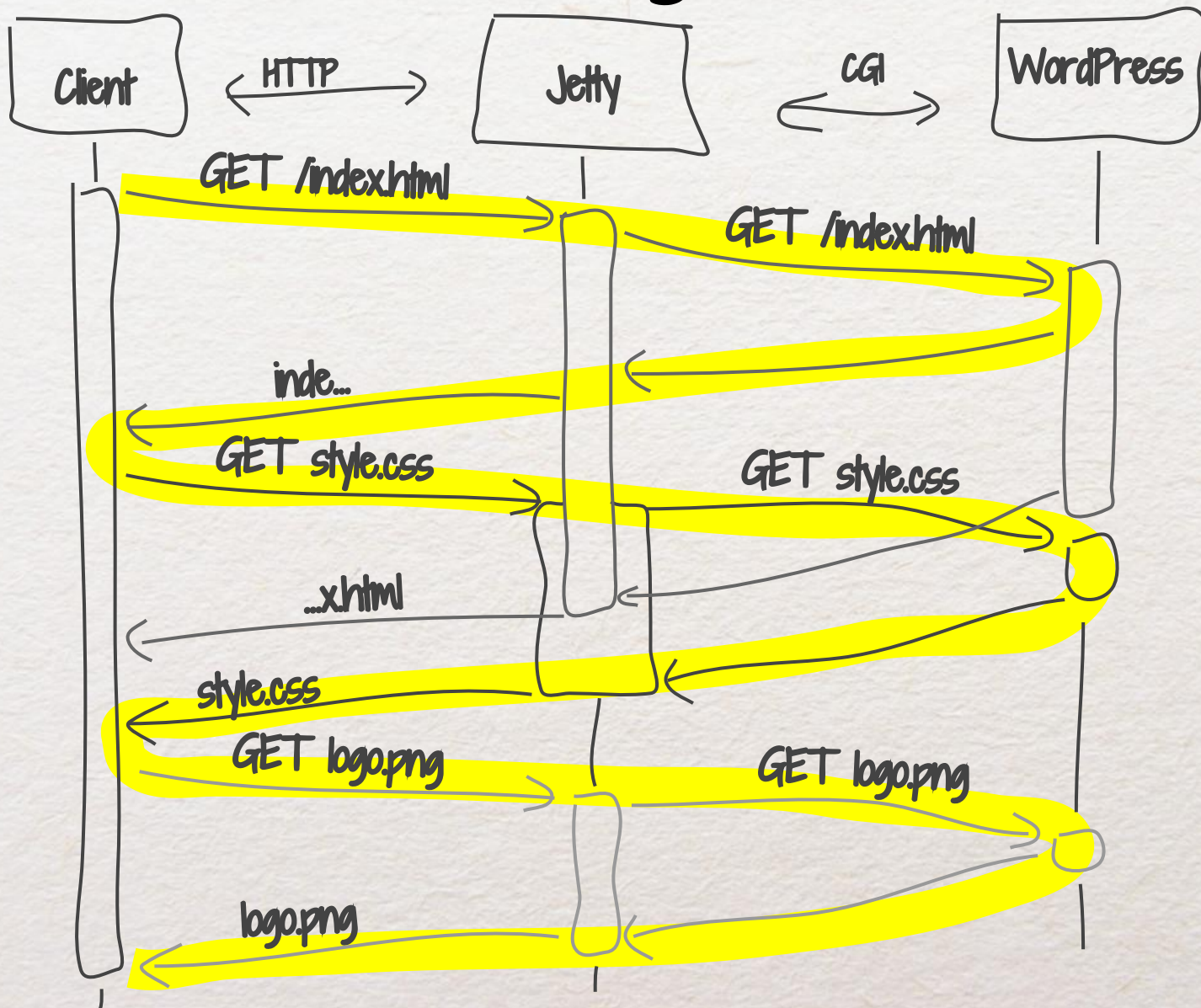
Iterative
when IO
is fast

Dispatched
when IO is
Slow

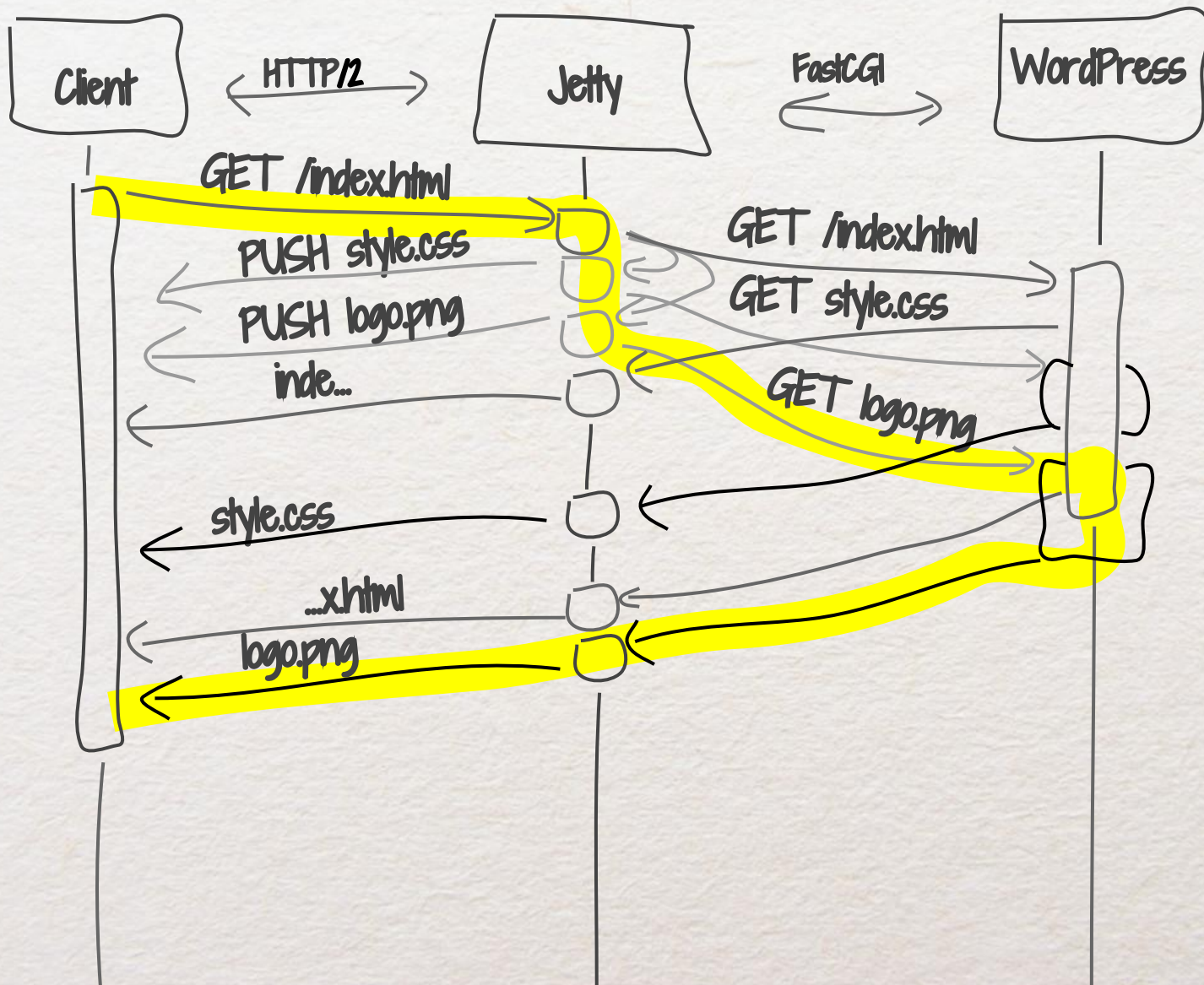
Async IO Demo

<http://webtide.com/http2-test/push>

HTTP and Blocking CGI



HTTP/2 and Async FastCGI



Async Only? PROS & CONS?

PROS

- All the wood behind one arrow!
 - code complexity
 - branch prediction
 - doco complexity
- Scaling
- Footprint

CONS

- No so good for
 - few very busy connections with simple requests
- Benchmark woes
 - which often use a few very busy connections with simple requests
- Usage is difficult
 - complex code
 - tough debug

Lies, Damned Lies & Benchmarks

- Most benchmarks
 - measure latency by reporting throughput
 - client do not offer more throughput often before server can't accept more
- Benchmarks should
 - Measure throughput by reporting latency!
 - Offer the throughput you want and report on QoS obtained
 - More subjective - eg long tails?

Jetty Modules

- Modularized Configuration of:
 - Classpath
 - XML files
 - Properties
 - Dependencies
 - Command line
- 3rd Party Downloads
 - Licences

Jetty Modules

```
$ java -jar $JETTY_HOME/start.jar --list-modules
```

```
...
```

```
- Module: jaas
```

```
  Depend: server
```

```
  LIB: lib/jetty-jaas-${jetty.version}.jar
```

```
  XML: etc/jetty-jaas.xml
```

```
  Enabled: <not enabled in this configuration>
```

```
...
```

```
$ java -jar $JETTY_HOME/start.jar --add-to-startd=jaas
```

```
$ java -jar $JETTY_HOME/start.jar --list-classpath
```

```
$ java -jar $JETTY_HOME/start.jar --list-config
```

```
$ java -jar $JETTY_HOME/start.jar --help
```


JAAS Module

```
$ cat $JETTY_HOME/modules/jaas.mod
```

```
# JAAS Module
```

```
[depend]
```

```
server
```

```
[lib]
```

```
lib/jetty-jaas-${jetty.version}.jar
```

```
[xml]
```

```
etc/jetty-jaas.xml
```

```
[ini-template]
```

```
## JAAS Configuration
```

```
jaas.login.conf=etc/login.conf
```


JAAS INI

```
$ cat start.d/jaas.ini
```

```
#
```

```
# Initialize module jaas
```

```
#
```

```
--module=jaas
```

```
## JAAS Configuration
```

```
jaas.login.conf=etc/login.conf
```


New Configuration Style

- Defined in `modules/name.mod`
- Enable modules `--add-to-startd`
 - creates `startd/name.ini`
 - references `name.xml`
- To change configuration:
 1. edit `startd/name.ini`
 2. edit `name.xml`
 3. create new `modules/name2.mod`

3rd Party Modules



ALPN RFC7301



Jetty BASE Directories

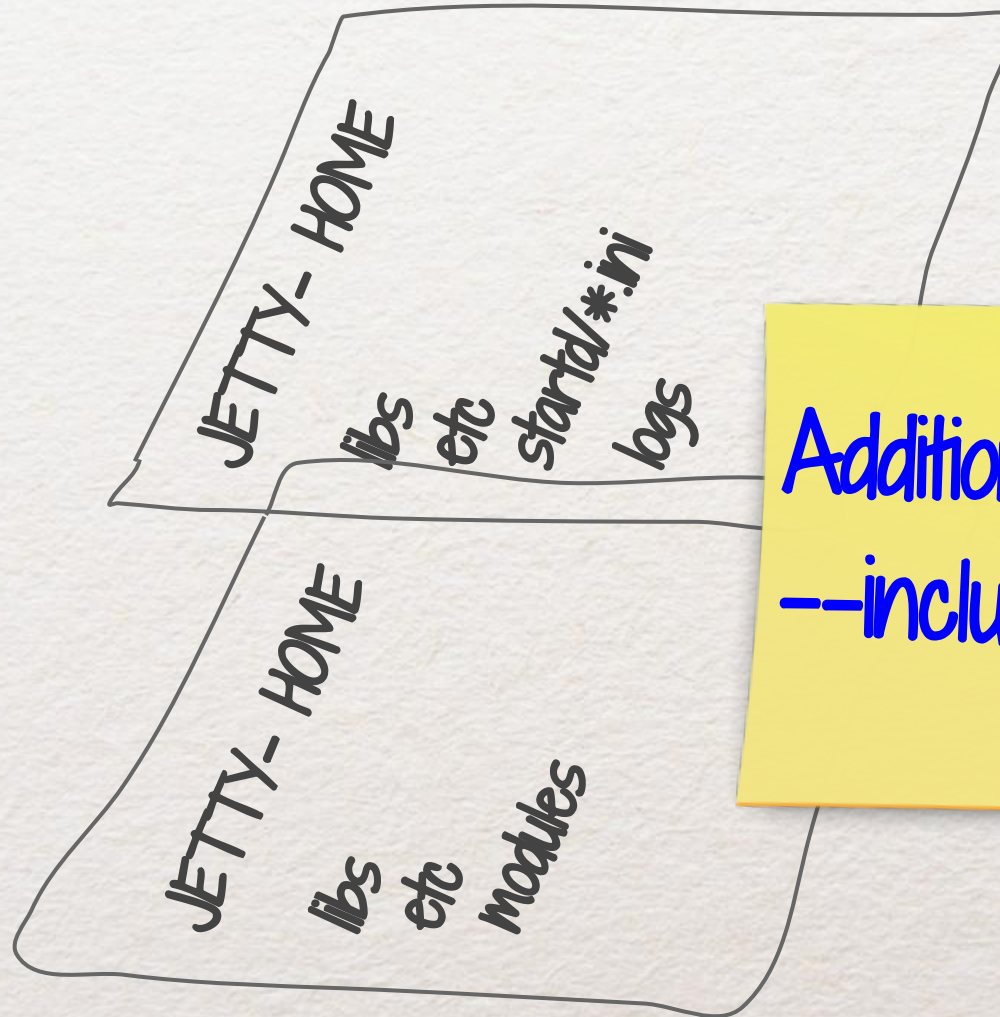
`${jetty.home}`

The Jetty Distribution (READONLY!)
defaults to start.jar directory

`${jetty.base}`

Layered ini,xml,lib,mod
defaults to .

Jetty Base & Home



Additional layers:
--include-jetty-dir

Jetty Base Demo

```
$ mkdir /tmp/demo
```

```
$ cd /tmp/demo
```

```
$ java -jar $JETTY_HOME/start.jar \  
--add-to-startd=http,deploy
```

```
$ cp $PROJECT/application.war webapps
```

```
$ java -jar $JETTY_HOME/start.jar
```


JEE Developer Assistance

Discover

Annotated Servlet

web.xml

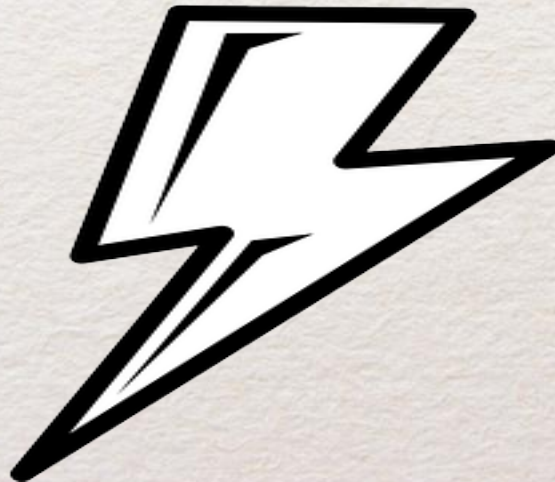
Servlet

Handler

TLD Descriptor

INF/classes

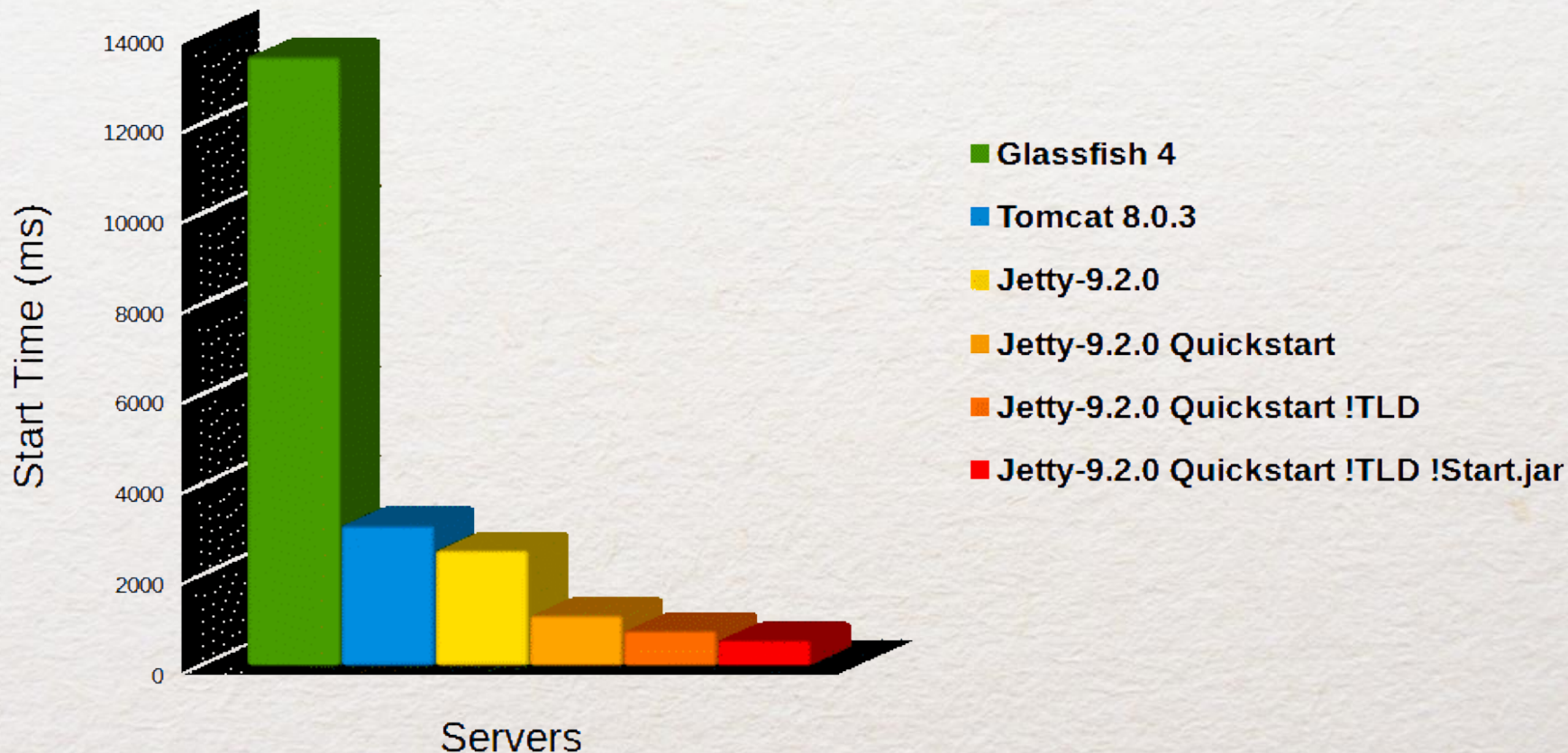
WEB-INF/lib



Quick Start Jetty

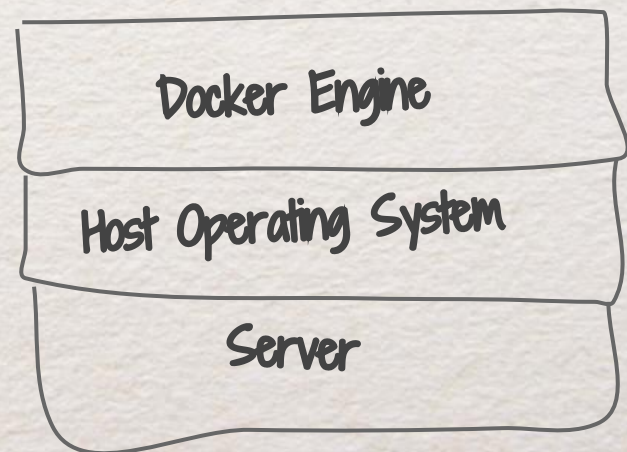
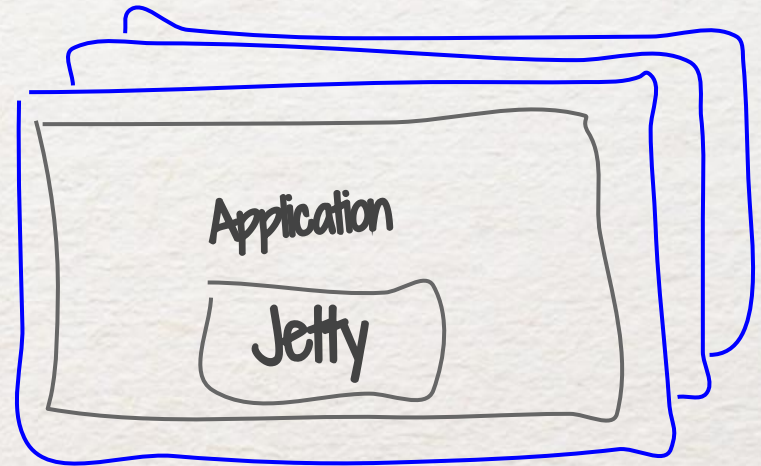
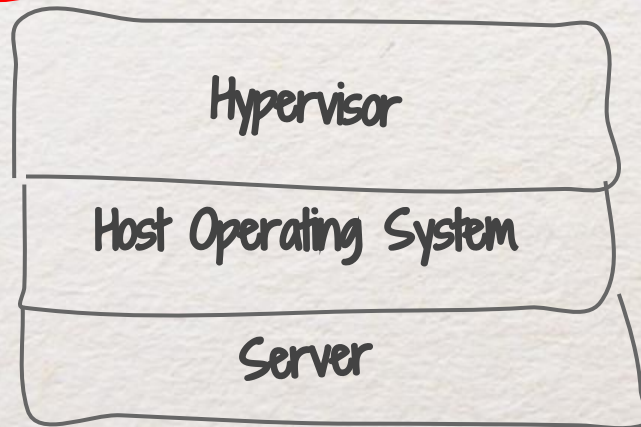
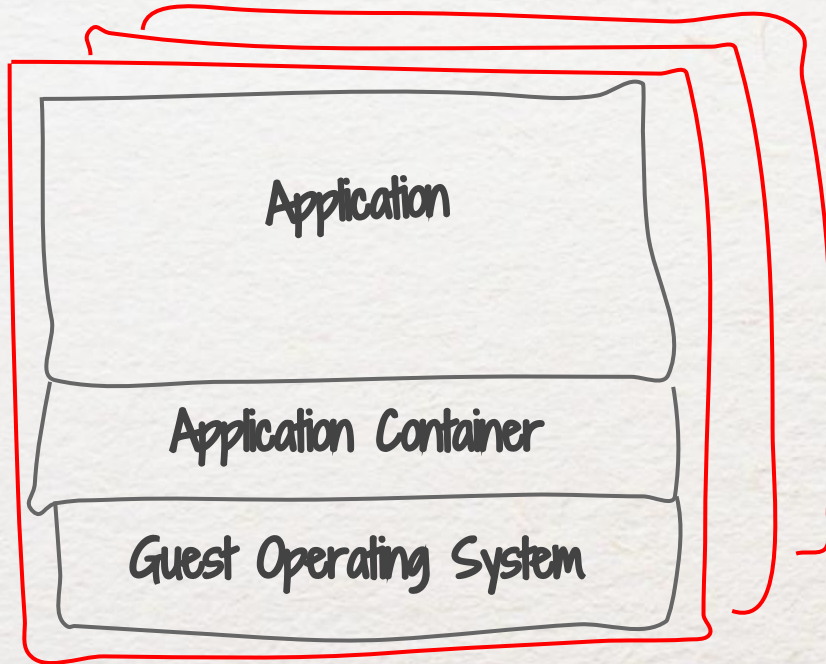
- Quick Start for the cloud:
 - Provision containers on demand
 - Save CPU on deploy
- QuickStart for security:
 - Disable all discovery:
 - Review Effective web.xml

Quickstart aiming for < 1000ms



Quick Start Demo

Great For Dockers!



Questions?

What other features are needed?

What Management Console?

Want HTTP/2 Test Sites!!!!!!

Greg Wilkins <gregw@webtide.com>

<http://www.webtide.com>