

DSLs, Scout and Docker in the Web

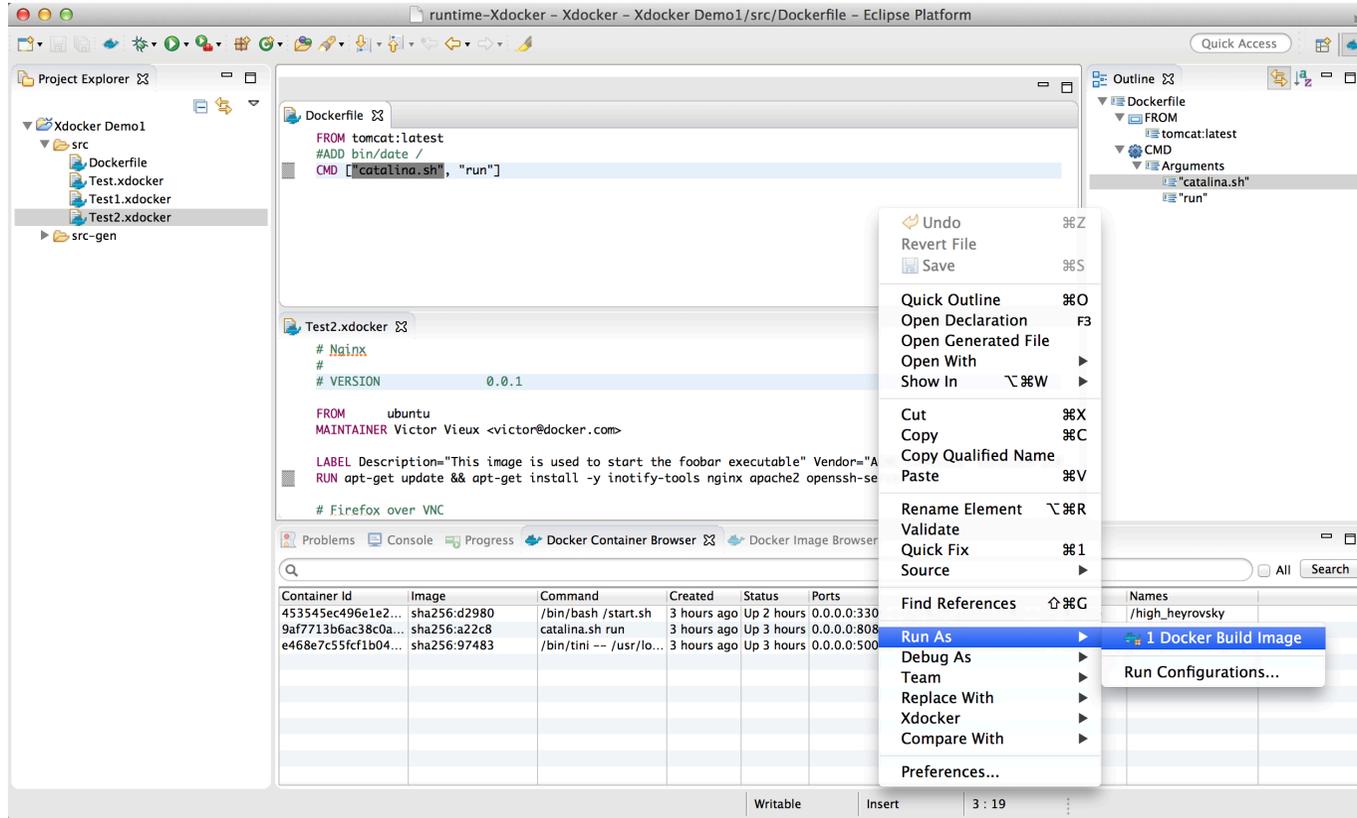
Eclipse DemoCamp – Zurich / 06-2016



Xdocker – An eXtensible Docker DSL with IDE Features



An Eclipse IDE for Docker written in Xtext, Xtend (90%) and Java



- Eclipse Features to manage Docker Files, Images, Container etc.
- Editor with Code Completion, Validation and Syntax Highlighting
- Build and run Docker Images with Launch Configuration

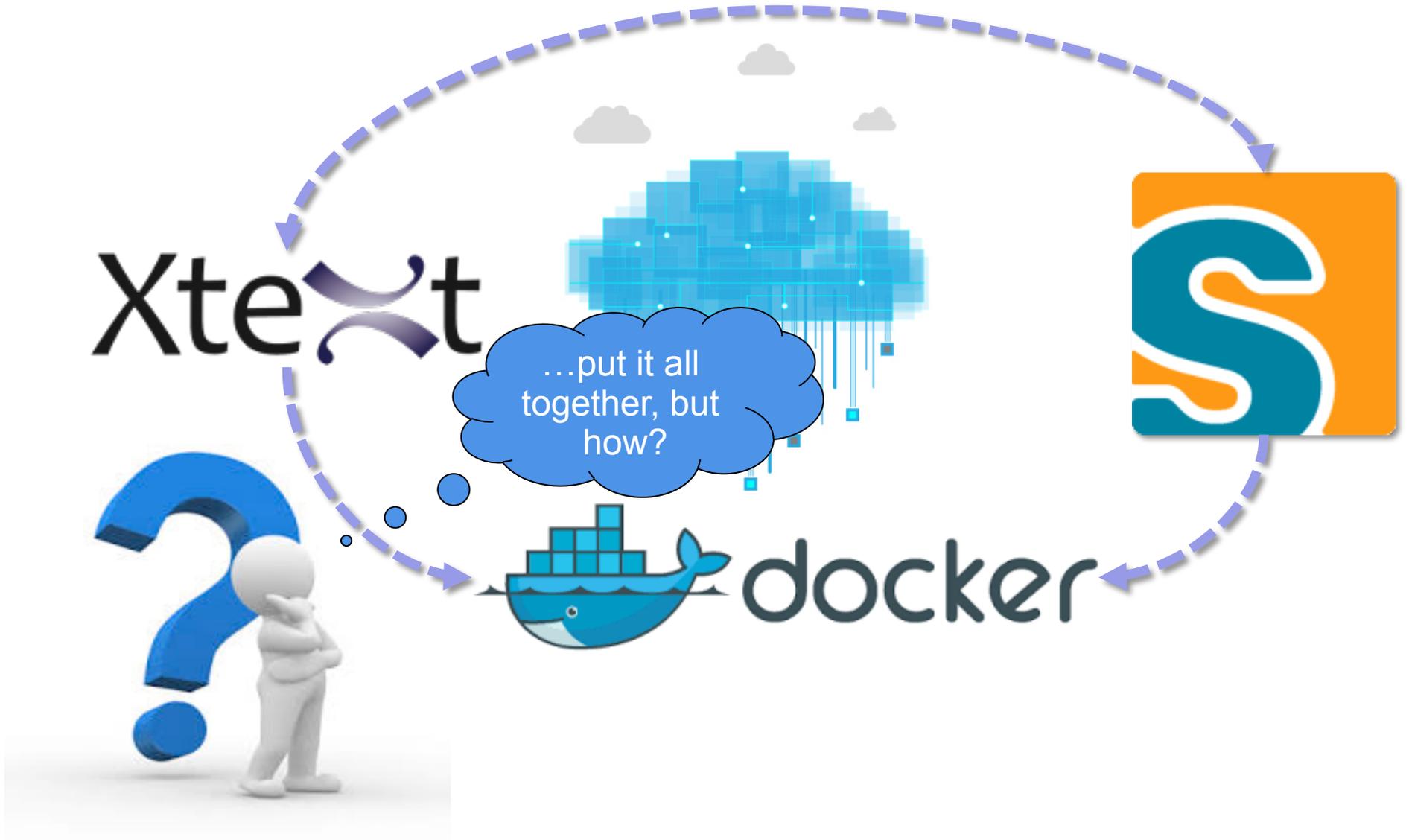
Xtext and Scout both support Web



Why not develop a Web DSL for Docker in a Scout Application?

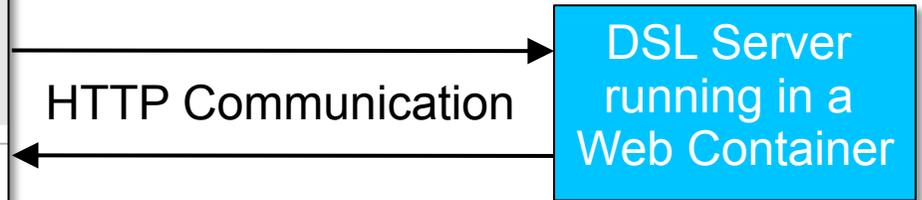
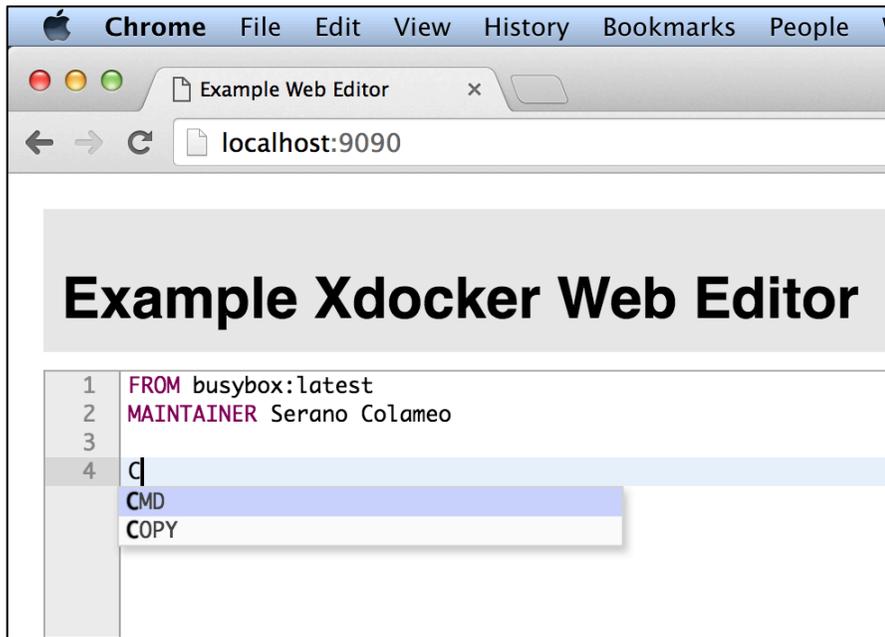


How to bring all these pieces together?



Xtext Web DSL Architecture

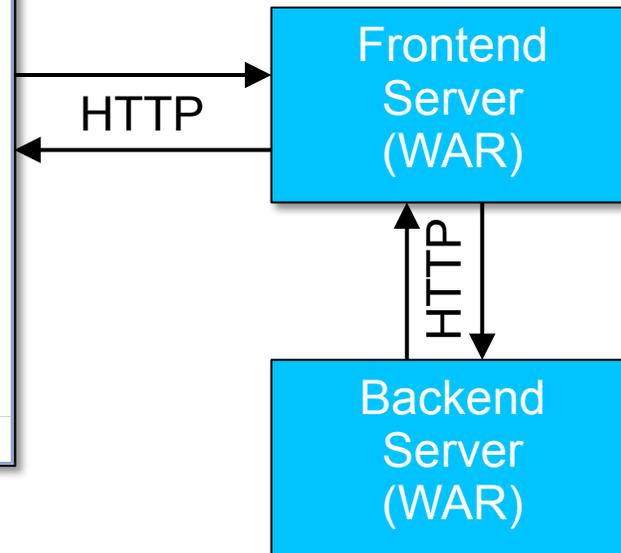
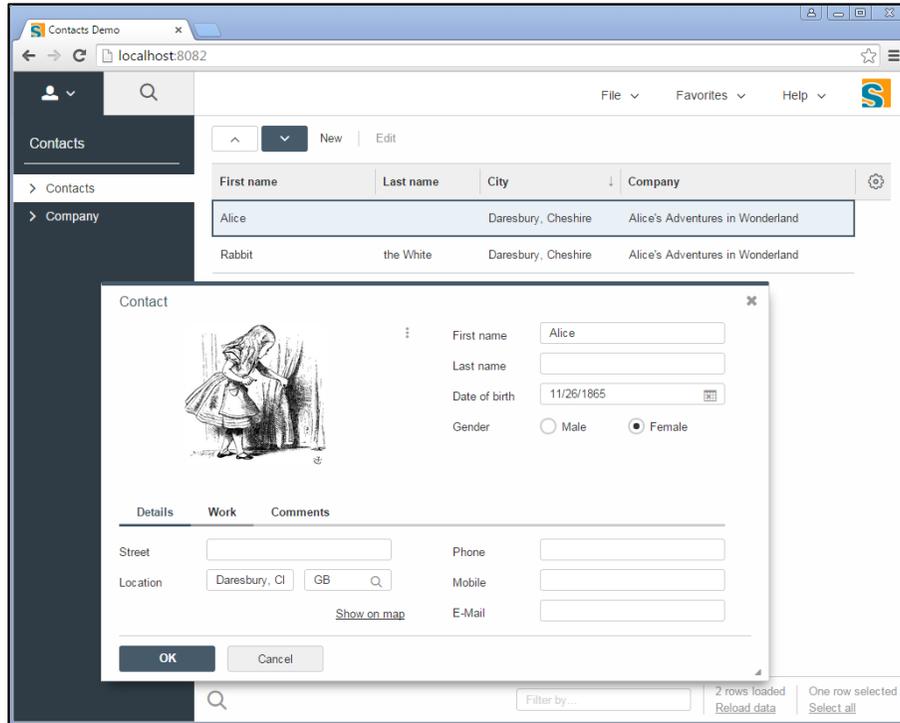
Migrate the Xdocker DSL to Xtext 2.10.x



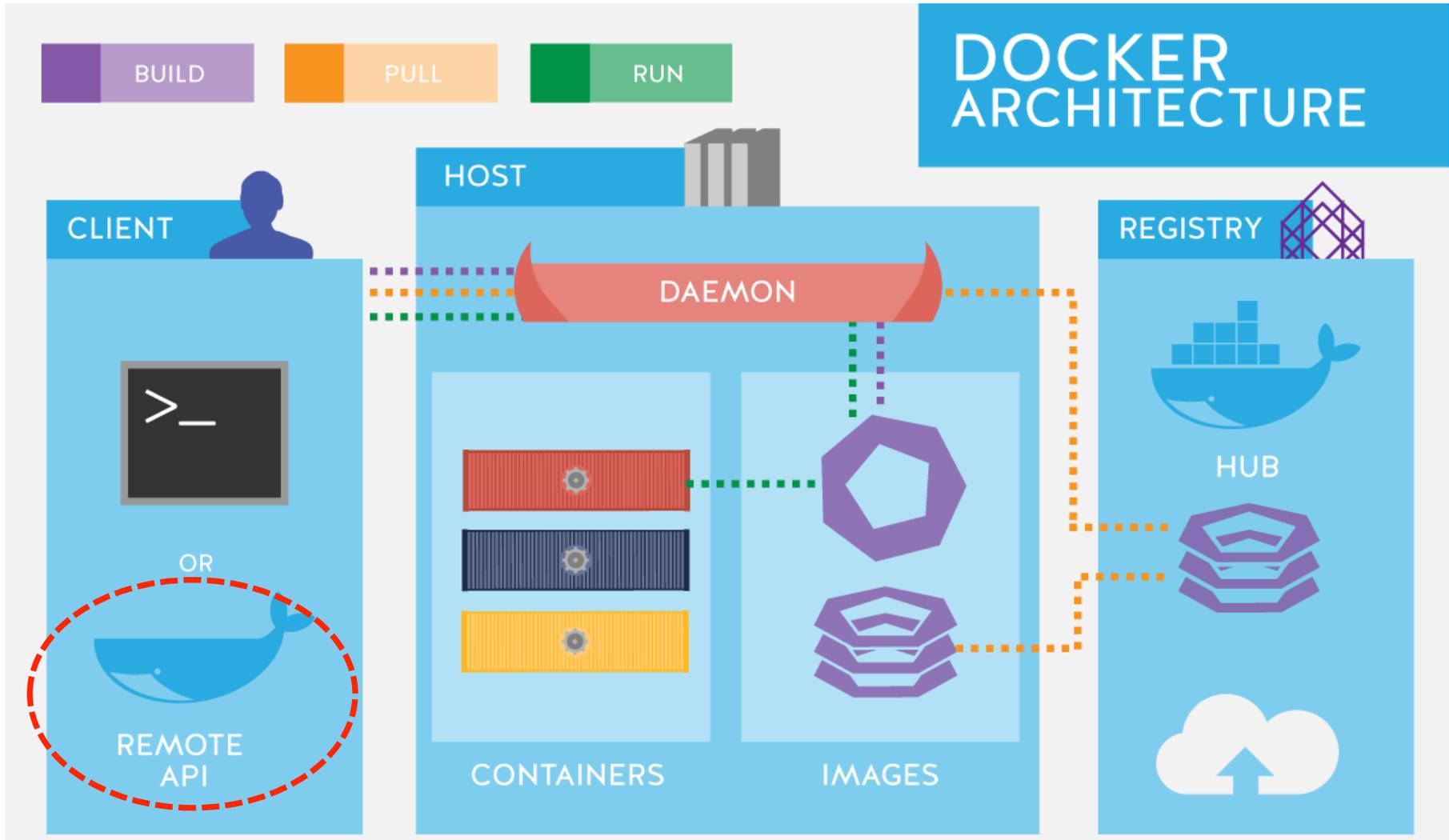
Scout Neon Architecture



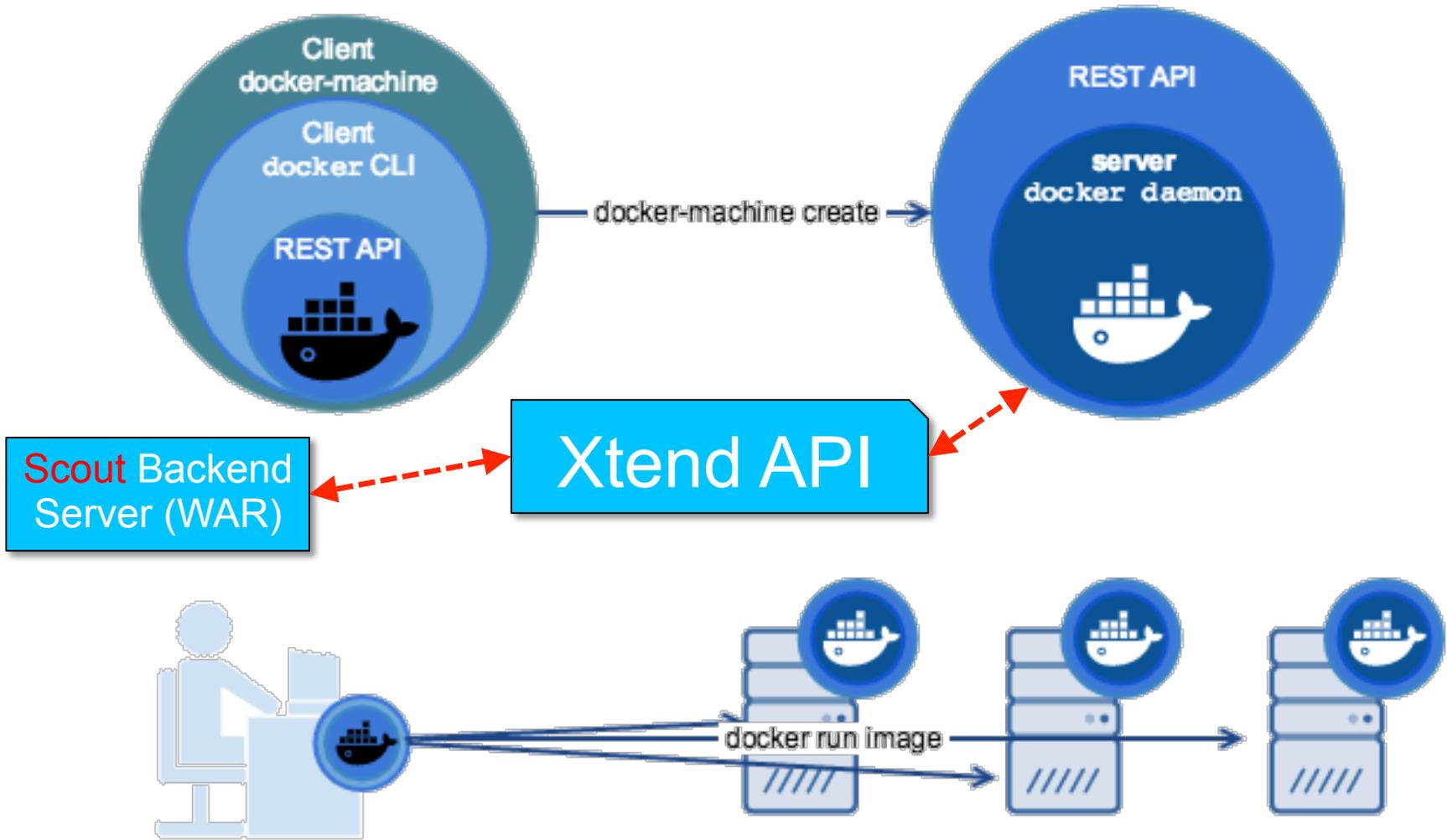
Understand how Scout Neon works, Web Integration, JavaScript, JQuery stuff etc.



Docker Architecture



Docker REST API Integration with Xtend



Xtend API for Docker as internal DSL



```
docker [  
    val callback = build('./tomcat8/docker')  
    val response = callback.awaitCompletion  
    println(response)  
  
    val id = image.run('/usr/share/tomcat8/bin/catalina.sh')  
    println(''Container id = «id»'')  
  
    images.forEach [  
        print(it.id + ' ')  
        println(it.repoTags.get(0))  
    ]  
  
    callback = pull('tomcat:latest')  
    val result = callback.awaitCompletion  
    println(result)  
    ...  
]
```

Xtend in Scout

```
115- override List<IImageInfo> search(String term, boolean isLocal) throws XdockerDockerServi
116     val List<IImageInfo> result = newArrayList
117     docker [
118         try {
119             if (isLocal) {
120                 images(term).forEach [
121                     result.add(new XdockerLocalImageInfo(repoTags, id, created, virtualS
122                 ]
123             } else {
124                 search(term).forEach [
125                     result.add(new XdockerRemoteImageInfo(name, description, starCount,
126                 ]
127             }
128         } catch (Exception ex) {
129             notify(ex.message, sessionId)
130         }
131     ]
132     return result
133 }
134
135- /**
136     * Build a dockerfile service operation
137     */
138- override build(String dockerfile) {
139     try {
140         Jobs.schedule(new IRunnable {
141             override run() {
142                 try {
143                     docker [
144                         val callback = new BuildImageResultCallback {
145                             override onNext(BuildResponseItem item) {
146                                 if (!item?.stream.isNullOrEmpty)
147                                     notify(item.stream, sessionId)
148                                 else if (!item?.status.isNullOrEmpty)
149                                     notify(''«item.status» «item.progress»'', sessionId)
150                                 else if (!item?.errorDetail?.message.isNullOrEmpty)
151                                     notify(item.errorDetail.message, sessionId)
152                                 else if (!item?.id.isNullOrEmpty)
153                                     notify(''id is «item.id»'', sessionId)
154                             }
155                         }
156-                     override onComplete() {
```



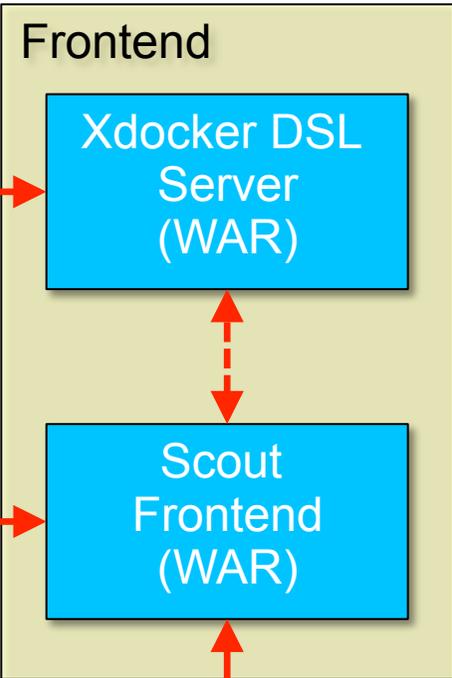
Xdocker Web App Architecture



```
Xdocker Scout Application
localhost:9082/#
Application Docker Help
Docker
New Open
Editor
Dockerfile
Save Close Build
1 FROM java:8-jdk
2
3 RUN apt-get update && apt-get install -y git curl zip && rm -rf /var/lib/apt/lists/*
4
5 MAINTAINER Serano Colameo
6
7 ENV JENKINS_HOME /var/jenkins_home
8 ENV JENKINS_SLAVE_AGENT_PORT 50000
9
10 ARG user=jenkins
11 ARG group=jenkins
12 ARG uid=1000
13 ARG gid=1000
14
15 # Jenkins is run with user `jenkins`, uid = 1000
16 # If you bind mount a volume from the host or a data container,
17 # ensure you use the same uid
18 RUN groupadd -g ${gid} ${group}
19     && useradd -d "$JENKINS_HOME" -u $uid -g $gid -m -s /bin/bash ${user}
20
21 # Jenkins home directory is a volume, so configuration and build history
22 # can be persisted and survive image upgrades
23 VOLUME /var/jenkins_home
24
Console
Clear Text
```

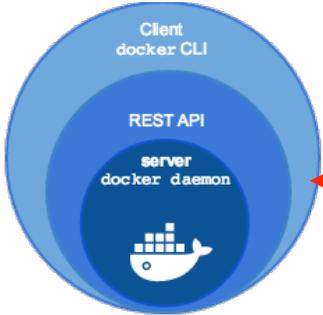
Scout/JS
Integration

Scout Client Model



Scout Backend Server (WAR)

Xtend API



Demo – „Xdocker in Action“





Serano Colameo

Phone: +41 (0)56 442 68 63

Email: serano.colameo@itemis.ch

itemis Schweiz GmbH | Dorfstrasse 69 | CH-5210 Windisch | www.itemis-schweiz.ch