

Enterprise Internet of Things

Dr. Mark Little, VP Engineering James Kirkland, Chief Architect, IoT David Ingham, Senior Engineering Manager

24th June 2015

Overview

- What is Enterprise Internet of Things?
 - Embedded, phones, cloud, Lego ...
 - Constrained devices?
- Applications and their requirements
- Red Hat and IoT
 - Intelligent Gateways
 - Interconnections (MQTT, AMQP, ...)
 - The developer option
- Where next?



The Internet of Things

- Explosive growth in connected devices
- Non-traditional interactions
 - Machine to machine (M2M)
 - Person to machine (P2M)
- Ubiquitous internet connectivity
- Affordable bandwidth
- Commoditized standardized hardware
 - Microcontrollers
 - Electronic sensors
- Advances in cloud computing
 - Decrease cost of storing and processing data
- Standards based and open source software
- Convergence of IT and OT





THE IOT INFORMATION LIFECYCLE



This work is liscensed under a Creative Commons Attriution-NonCommercial-ShareAlike 4.0 International License



Why Enterprise IoT Needs 3 Tiers



Transmission costs paid by consumer



Smaller amounts of data



Analysis and response not critical



Bandwidth and network transports not a concern



Simple network communications



No need for data summarization



A 3-tier Architecture for IoT







sp. redhat



Device tier Millions of instances

RH0049-2C



Control tier Thousands of instances





Datacenter/cloud tier Hundreds of instances



Identity Management Red Hat CloudForms

OpenShift by Red Hat



* provided by Red Hat certified partners

IoT Messaging Patterns



Telemetry 1-way, device to service



Command & Control Request/reply, service to device



Enquiry Request/reply, device to service



Notification 1-way, service to device



IoT Connectivity Patterns







Protocols: MQTT

- Connection-oriented, bi-directional, binary, broker-based, TCP protocol
- Suitable for constrained devices and low-bandwidth, high-latency networks
- Binary message payload, no structured message format
- Range of quality-of-service guarantees
- SSL tunnelling for security
- An OASIS Standard; mature, popular, and widely-supported





Protocols: AMQP 1.0



- Initially created by JP Morgan and industry consortium; now an International Standard (ISO/IEC ISO 19464); mature, popular, and widely-supported
- Connection-oriented, bi-directional, binary TCP protocol
- Flexible deployment: broker-based or peer-to-peer
- Rich feature set: conversation multiplexing, advanced flow control, ...
- No broker messaging model defined; reliable exchange between 2 endpoints
- Includes structured message format which facilitates interoperability
- Range of quality-of-service guarantees
- SSL tunnelling for security



Comparing protocols

- Lots of IoT protocol choices including several not discussed here
- The right choice depends of the scenario
- Scenarios may combine protocols

	MQTT	AMQP	СоАР	STOMP
Target deployment	Long-haul (& local)	Long-haul	Local (& long-haul)	Long-haul
Compactness	High	Medium	Highest	Low
Security	SSL	SSL	DTLS	SSL
Flow control	No	Yes	No	No
Structured message	No	Yes	No	Yes
Multiplexed conversations	No	Yes	No	No



Connecting the IoT with A-MQ

- Red Hat JBoss A-MQ is a high-performance, scalable messaging broker
- Works great as a server-side broker for connecting IoT devices
- Can be used with Red Hat JBoss Fuse to add transformation and routing capabilities
- Supports AMQP 1.0, MQTT 3.1.1, HTTP, STOMP protocols
- Broad range of supported client libraries for all kinds of device hardware





Example: Delegate Tracking







Developers and IoT

- Why spend the time?
 - Non-traditional enterprise deployments
 - These demands are coming ...
- Focus on MBed, Pi, Mobile (Android)
 - Not just edge devices
 - http://www.jboss.org/iot/
- Looking at core capabilities initially
 - Well ... maybe a bit of fun too
 - Narayana, Arquillian, Vert.x, ActiveMQ, Infinispan, Fabric, jBPM, ...



A-MQ in space!





Apollo and High Energy Physics





Java not up to the task?

- JavaScript and Node.js (client and server)
- Go[lang] (Cloud, Docker, Kubernetes)
- Erlang (enterprise messaging)
- Dart (client and server)
- Swift (mobile)
- Objective-C (mobile)
- Python (Cloud, OpenStack)
- C/C++ (client/edge-devices and server)





Why should Java developers be interested?

- Types of IoT application increasing in complexity
 - Online purchases
 - Distributed peer-to-peer interactions
- More requirements becoming a necessity
 - Security and identity cloud breaches bad enough!
 - High performance, low latency, reliable messaging
 - Database updates with transactions
 - Workflows as inter-app interactions increase



Familiar requirements?

- Many commonalities with Java middleware
 - Messaging
 - Transactions
 - Security
- Non-JVM languages are a necessity, e.g., clients on "thin" (edge?) devices
- But we cannot afford to re-implement the world
 - Java has a significant role to play



Java 2000 versus 2015

- J2SE 1.3
- J2EE 1.2 to 1.3
- Typical laptop configuration
 - 1 GHz Pentium III with
 512 Meg and 20 Gig disk
 - Sufficient to run full J2EE stack
- Limited mobile devices
 - HP Jornada 720, 206MHz
 StrongArm SA1110 32-bit
 processor, 32 MB SDRAM,
 Windows CE
 - Sufficient to run some components of J2EE

- Java SE 8
- Java EE 6/7
- Typical laptop configuration
 - 2.5 GHz Quad Core i7 with
 16 Gig and 256 Gig SSD
 - Sufficient to run full EE6 stack (on several different OS concurrently)
- Smartphones
 - Samsung Galaxy S5, Quad Core 2.5 GHz Krait 400
 - Sufficient to run many components of EE6







JBOSS_HOME: /home/jboss/jboss-as-7.1.0.CR1b

JAVA: java

JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=360 000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true

10:45:31,254 INFO [org.jboss.modules] JBoss Modules version 1.1.0.CR6 10:45:33,065 INFO [org.jboss.msc] JBoss MSC version 1.0.1.GA 10:45:33,567 INFO [org.jboss.as] JBoss AS 7.1.0.CR1b "Flux Capacitor" starting 10:45:40,621 INFO [org.jboss.as] Creating http management service using socket-binding (management-http) 10:45:40,632 INFO [org.xnio] XNIO Version 3.0.0.CR7 10:45:40,709 INFO [org.xnio.nio] XNIO NIO Implementation Version 3.0.0.CR7 10:45:40,709 INFO 10:45:40,753 INFO [org.jboss.remoting] JBoss Remoting version 3.2.0.CR8 10:45:41,460 INFO [org.]boss.as.logging] JBAS011502: Removing bootstrap log handlers 10:45:41,703 INFO [org.jboss.as.naming] (ServerService Thread Pool -- 38) JBAS011800: Activating Naming Subsystem 10:45:41,739 INFO [org.jboss.as.clustering] (ServerService Thread Pool -- 30) JBAS010300: Activating Infinispan subsystem. 10:45:41,753 INFO [org.jboss.as.security] (ServerService Thread Pool -- 44) Activating Security Subsystem 10:45:41,912 INFO [org.]boss.as.osgi] (ServerService Thread Pool -- 39) JBAS011910: Activating OSGi Subsystem 10:45:41,942 INFO [org.jboss.as.security] (MSC service thread 1-1) Picketbox version=4.0.6.Beta2 10:45:41,967 INFO [org.]boss.as.mebservices] (ServerService Thread Pool -- 48) JBAS015537: Activating WebServices Extension 10:45:42,236 INFO [org.]boss.as.connector] (MSC service thread 1-4) JBAS010408: Starting JCA Subsystem (JBoss IronJacamar 1.0.6.Final) 10:45:42,759 INFO [org.]boss.as.naming] (MSC service thread 1-4) JBAS011802: Starting Naming Service 10:45:42,865 INFO [org.]boss.as.jaxr] (MSC service thread 1-4) Binding JAXR ConnectionFactory: java:jboss/jaxr/ConnectionFactory 10:45:43,066 INFO [org.]boss.as.mail.extension] (MSC service thread 1-2) JBAS015400: Bound mail session [java:jboss/mail/Default] 10:45:44,035 INFO [org.jboss.as.connector.subsystems.datasources] (ServerService Thread Pool -- 26) JBAS010403: Deploying JDBC-compliant driver class org.h2.Dr iver (version 1.3) 10:45:44,116 INFO [org.jboss.as.remoting] (MSC service thread 1-1) Listening on /127.0.0.1:4447 10:45:44,378 INFO [org.apache.catalina.core.AprLifecycleListener] (MSC service thread 1-3) The Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: /usr/lib/jvm/java-6-openjdk/jre/lib/arm/server:/usr/lib/jvm/java-6-openjdk/jre/lib/arm:/usr/lib /jwm/java-6-openjdk/jre/../lib/arm:/usr/java/o-openjdk/jre/lib/arm:/usr/lib/ /jwm/java-6-openjdk/jre/../lib/arm:/usr/java/o-openjdk/jre/lib/arm:/usr/lib/ 10:45:45,549 INFO [org.apache.coyote.htp11.Htp11Protocol] (MSC service thread 1-1) Starting Coyote HTTP/1.1 on http--127.0.0.1-8080 10:45:47,228 INFO [org.jboss.ws.common.management.AbstractServerConfig] (MSC service thread 1-2) JBoss Web Services - Stack CXF Server 4.0.0.GA 10:45:48,434 INFO [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-1) JBAS010400: Bound data source [java:jboss/datasources/ExampleDS] 10:45:49,275 INFO [org.jboss.as.remoting] (MSC service thread 1-2) Listening on /127.0.0.1:9999 10:45:49,359 INFO [org.jboss.as.server.deployment.scanner] (MSC service thread 1-1) JBA5015012: Started FileSystemDeploymentService for directory /home/jboss/j boss-as-7.1.0.CR1b/standalone/deployments 10:45:49,955 INFO [org.jboss.as] (Controller Boot Thread) JBoss AS 7.1.0.CR1b "Flux Capacitor" started in 20090ms - Started 130 of 200 services (68 services ar passive or on-demand)







Transactional Android

```
public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
TextView tv = new TextView(this);
```

```
AtomicAction A = new AtomicAction();
AtomicObject foo = new AtomicObject();
```

```
A.begin();
```

```
foo.set(2);
```

```
A.commit();
```

}

}

```
tv.setText("Hello, TxAndroid created "+A.toString()+" and got "+foo.get());
setContentView(tv);
```

Code changes?

- For Raspberry Pi?
 - It's standard Java
 - No code changes
 - Just add a little more patience
- For Android?
 - No StAX by default
 - Classloading is *very* different
 - Remember to use —core-library to get javax
 - Some pre-JDK 1.6 compilation issues ...



Conclusions

- IoT applications need some capabilities normally associated with enterprises
 - Next generation IoT applications will need more
 - Trillions of sensors
 - Billions of phones
- IoT starts and ends with communication
 - MQTT, AMQP
- Gateways (intelligent or not) critical to scale and reliability
- Red Hat has a lot to offer developers today and in the future







THANK YOU



You Tube plus.google.com/+RedHat

linkedin.com/company/red-hat

voutube.com/user/RedHatVideos



f

twitter.com/RedHatNews

facebook.com/redhatinc

32