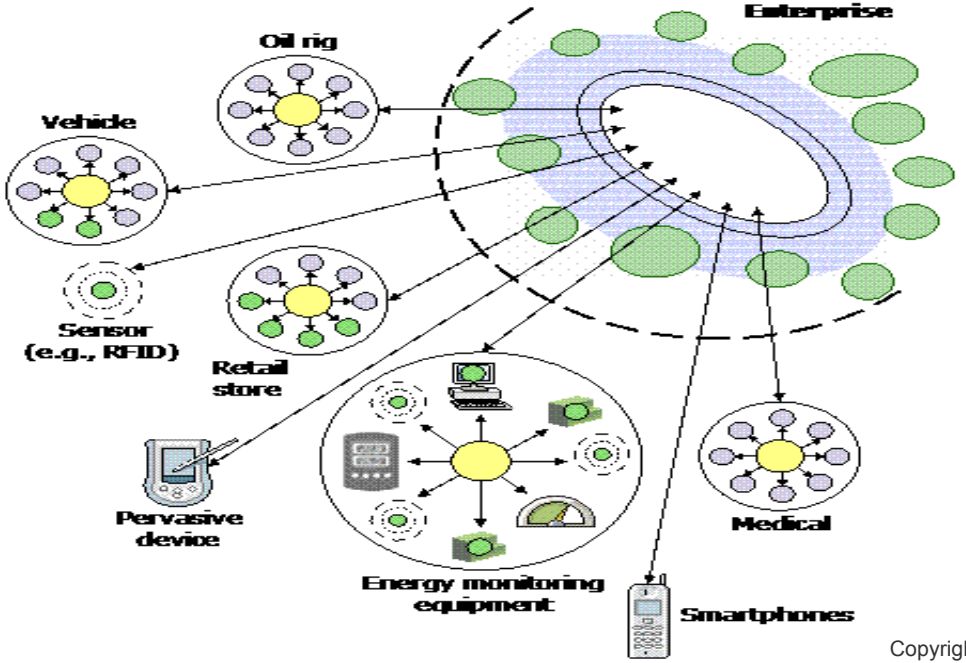
# Spring Integration Done Bootifully

**Glenn Renfro**, Pivotal
@cppwfs

MQTT.ORG








spring

# Ooonnnneeee Last Minute Change




spring

# Spring Boot

Takes an opinionated view of building production-ready Spring applications. Spring Boot favors convention over configuration and is designed to get you up and running as quickly as possible.

- Create stand-alone Spring applications
- Embed Tomcat or Jetty directly (no need to deploy WAR files)
- Opinionated 'starter' POMs
- Automatically configure Spring whenever possible
- Absolutely no code generation and no requirement for XML configuration

# Spring Integration

Extends the Spring programming model to support the well-known Enterprise Integration Patterns.

- Spring Integration enables lightweight messaging and supports integration with external systems.
- Adapters provide a higher-level of abstraction over Spring's support for remoting, messaging, and scheduling.
  - ReST/HTTP
  - SFTP/FTP
  - RabbitMQ
  - JMS
  - TCP/UDP
- Spring Integration's primary goal is to provide a simple model for building enterprise integration solutions while maintaining the separation of concerns that is essential for producing maintainable, testable code.

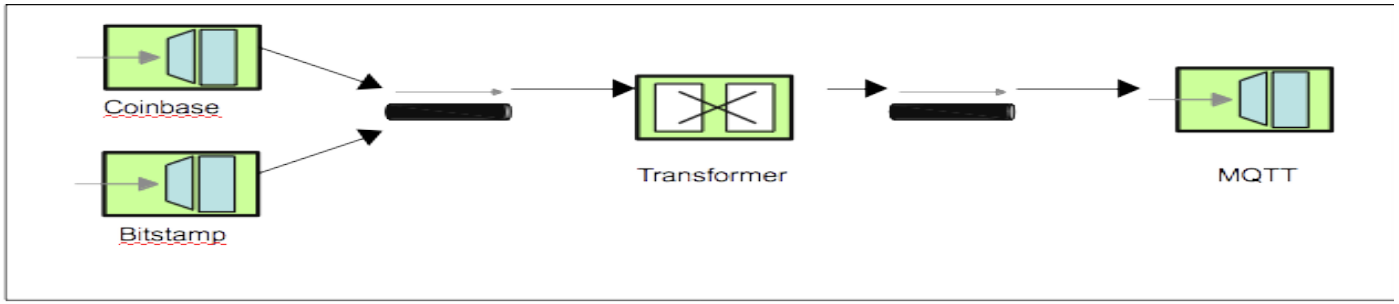


spring

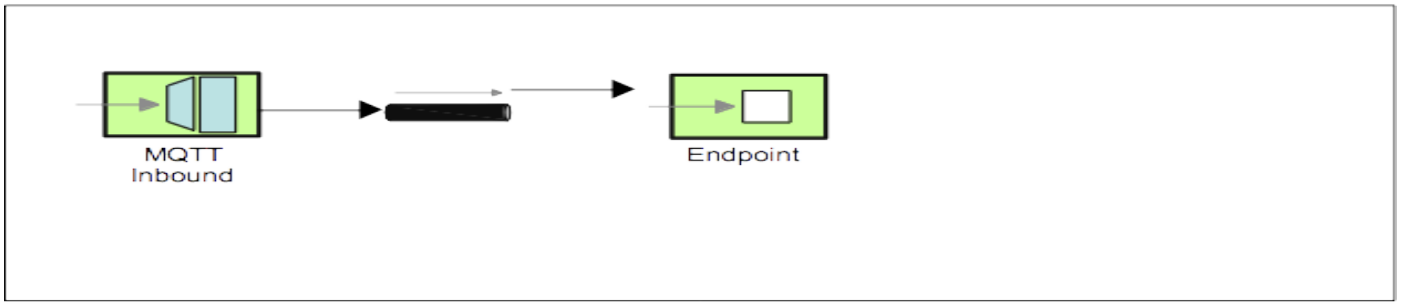
# Agenda

- **Create a SI application**
  - that gathers bid data from various bitcoin banks and markets.
  - Translate data
  - Send data MQTT Client
- **Create MQTT Client**

# Data Flow




Bitcoin Monitor




MQTT Monitor



spring

# The Tools

- Your Favorite Editor
- Gradlew
- Git (Optional)
- RabbitMQ Need port 1883.

# The Libraries

- Spring Boot
- Spring Integration
- Jackson

# The Gradle File

```
buildscript {
    repositories {
        maven { url "http://repo.spring.io/libs-snapshot" }  mavenLocal() }
        dependencies {
            classpath("org.springframework.boot:spring-boot-gradle-plugin:1.0.0.RC3”) }
}
…
apply plugin: 'spring-boot'
…

dependencies {
                compile("org.springframework.boot:spring-boot-starter-integration")
                compile("com.fasterxml.jackson.core:jackson-databind")
                testCompile("junit:junit")
}
```

# Get the Project

- If using git
  - Cd to or create a temporary directory
  - git clone git@github.com:cppwfs/bitcoinmonitor.git
  - git clone git@github.com:cppwfs/mqttmonitor.git
- Else
  - Copy the BitCoinMonitor project to your computer from the thumb drive to a temp dir
- Build the bitcoin project
  - cd bitcoinmonitor
  - ./gradlew build –refresh-dependencies
- Instructions are located at: https://github.com/cppwfs/bitcoinmonitor/blob/master/README.md