WebSocket Perspectives 2015

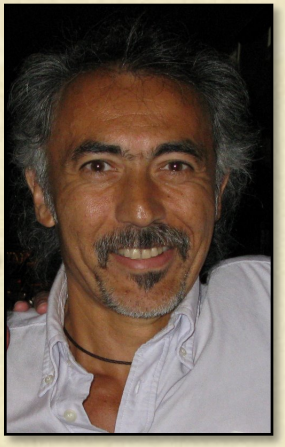
Clouds, Streaming, Microservices and the Web of Things

@frankgreco and @pmoskovi

goo.gl/ebiksb

KAAZING

Background



- Director of Technology
- Chairman NYJavaSIG (javasig.com)
- Largest Java UG in NA 8k+ members
- Chair NYHTML5 1,500+ members
- frank.greco@kaazing.com, @frankgreco

- Head of Real-Time Solutions
- Co-author Definitive Guide to WebSocket
- Co-author Oracle WebCenter 11g Handbook
- peter.moskovits@kaazing.com, @pmoskovi

WIN A COPY!

goo.gl/ebiksb

KAAZING

WIN A COPY!

1. Introduction to HTML5 WebSocket
2. The WebSocket API
3. The WebSocket Protocol
4. Building Instant Messaging and Chat over WebSocket with XMPP
5. Using Messaging over WebSocket with STOMP
6. VNC with the Remote Frame Buffer Protocol
7. WebSocket Security
8. Deployment Considerations

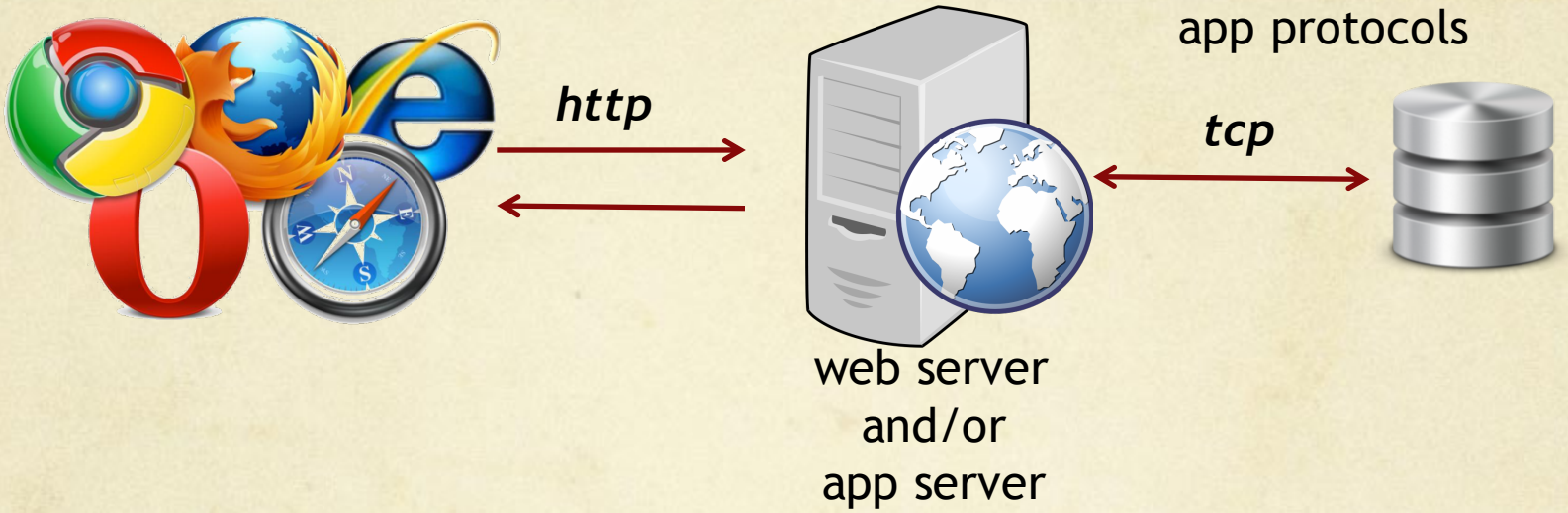
goo.gl/ebiksb

Outline – Things to Consider

- Web Communications Then and Now
- Web APIs
- Communications Models, Protocols, Frameworks
- Where WebSocket Fits
  - IoT/WoT
  - Microservices Transports
  - Cloud Connectivity

Web Communication

Web – “over the firewall” (early 90’s – 2011)



Page and Visitor hits used to be the report card

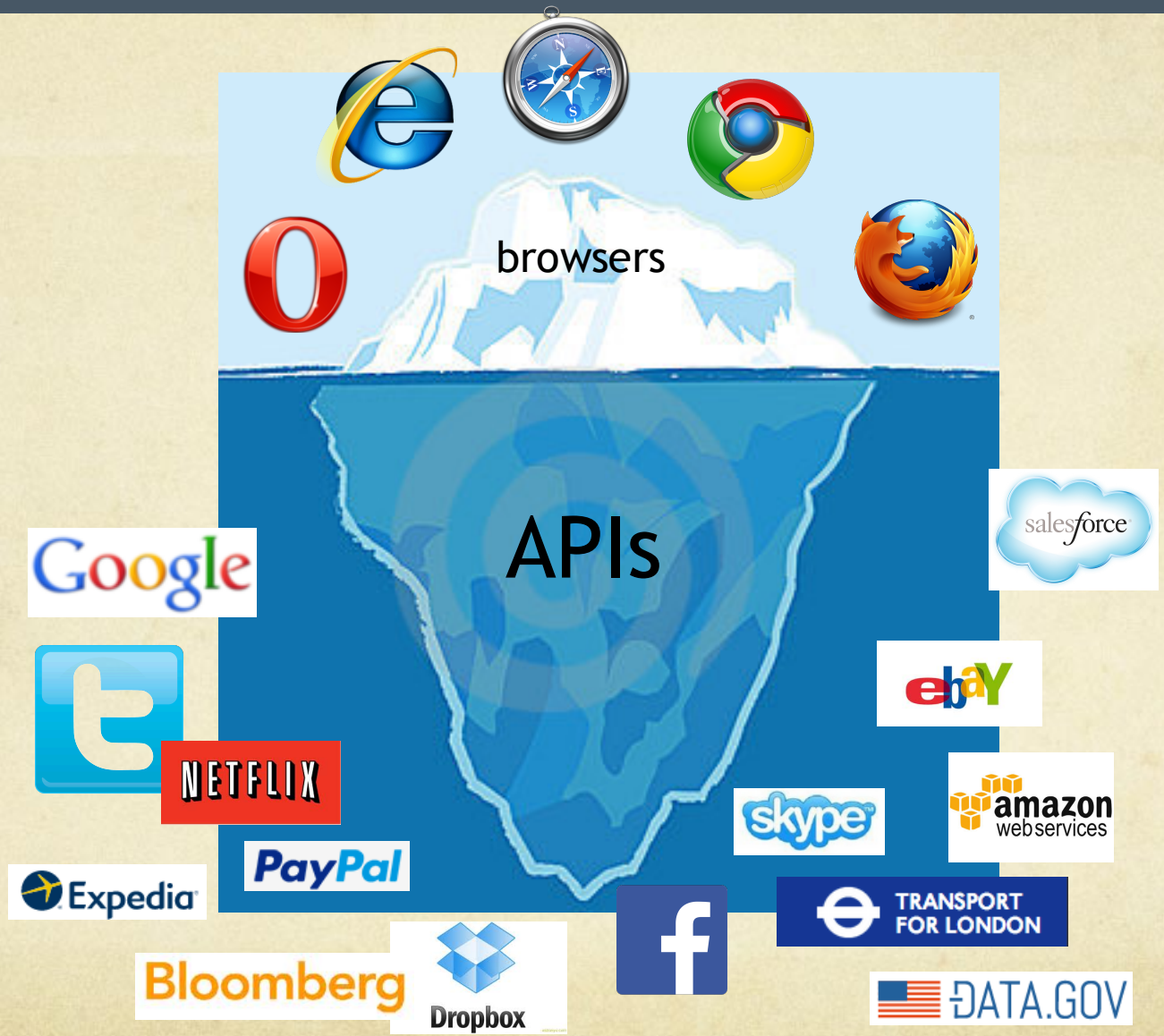
You are visitor 001234 ←Remember these?

goo.gl/ebiksb

KAAZING

Web APIs

The Hidden Web – Most of the Web is Not Visible



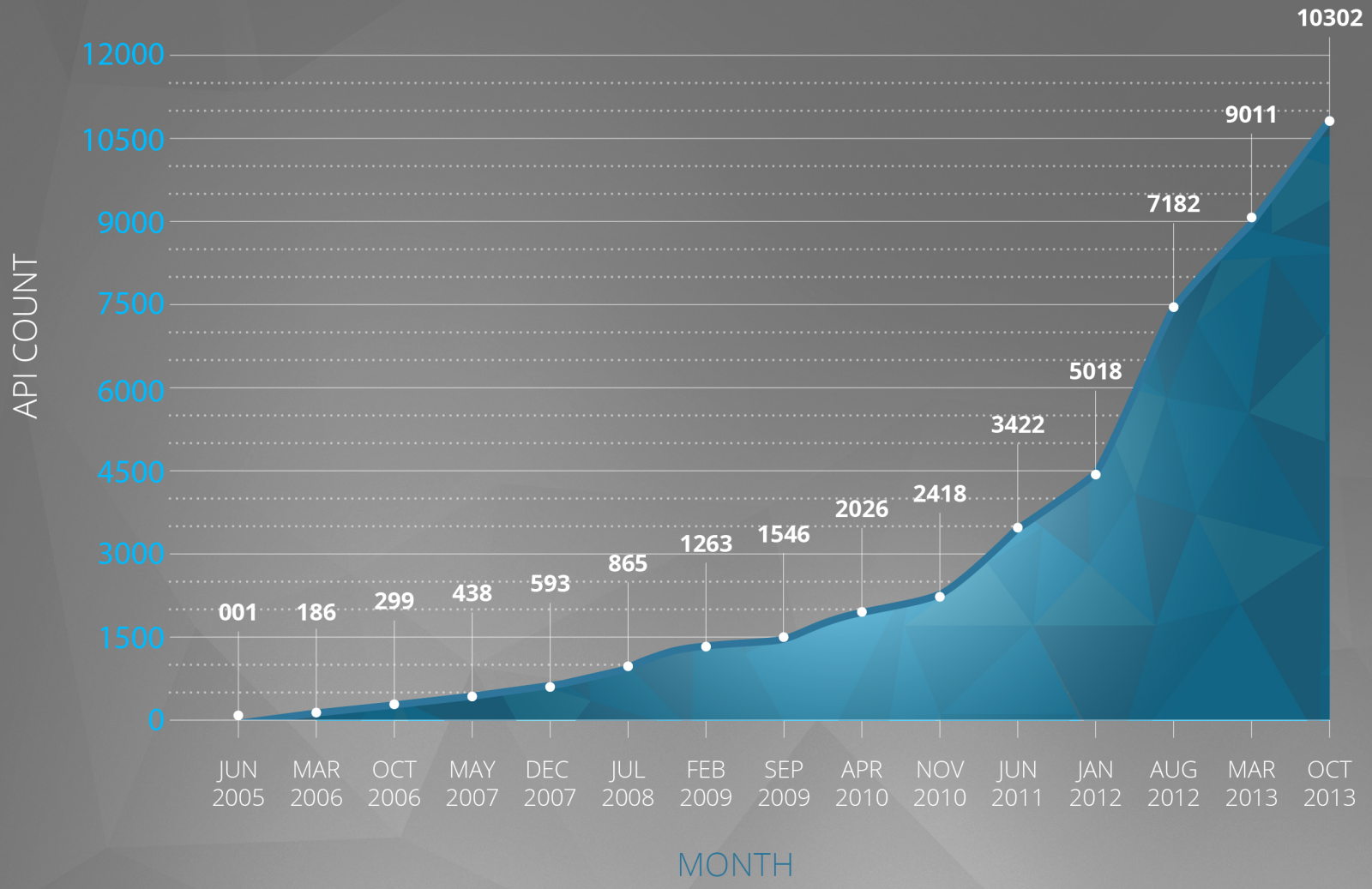
http://thumbs.dreamstime.com/x/iceberg-23503494.jpg

goo.gl/ebiksb

KAAZING

ProgrammableWeb

Growth In Web APIs Since 2005



2015

You are API call # 3,617,293,229 ***today***

2012

You are visitor # 2,391 this year

goo.gl/ebiksb

KAAZING

Explosion of Open Web APIs

- APIs from Everywhere, Consumed by Every [one|thing]
- ~14K public APIs and even more Mashups
  - programmableweb.com/apis/directory
  - Amazon, Facebook, LinkedIn, AT&T, Google, Microsoft, NYTimes, Orange, SalesForce, Telefonica, Twitter, Visa, Vodafone, Bloomberg, NYSE, Thomson-Reuters, etc.
- Over time, more will be event-based
- Enterprise and B2B APIs
- Services… Services… Services…

programmableweb

API FOR THAT

DATA.GOV

PublicAPIs
by Mashape API Platform

Chuck Norris has an API and It can Kill you

```
% curl http://api.icndb.com/jokes/random 2>&1 | grep joke |\
        tr -d "{}" | sed -e 's/.*joke": "//' -e 's/".*$//'
```

Chuck Norris can overflow your stack just by looking at it.


```
% curl http://api.icndb.com/jokes/random 2>&1 | grep joke |\
        tr -d "{}" | sed -e 's/.*joke": "//' -e 's/".*$//'
```


Chuck Norris is the only man who has, literally, beaten the odds. With his fists.

Using the Web without a Browser

```
% REPO=kaazing/gateway

% printf 'As of %s, repo [%s] has %s forks\n' \
    "`date +%D`" \
    $REPO \
    `curl --user "XXXXX:YYYYY" https://api.github.com/repos/$REPO 2>&1 \
    grep -i forks_count | \
    cut -d: -f2 | \
    tr -d ,`
```

As of 10/16/15, repo [kaazing/gateway] has 34 forks

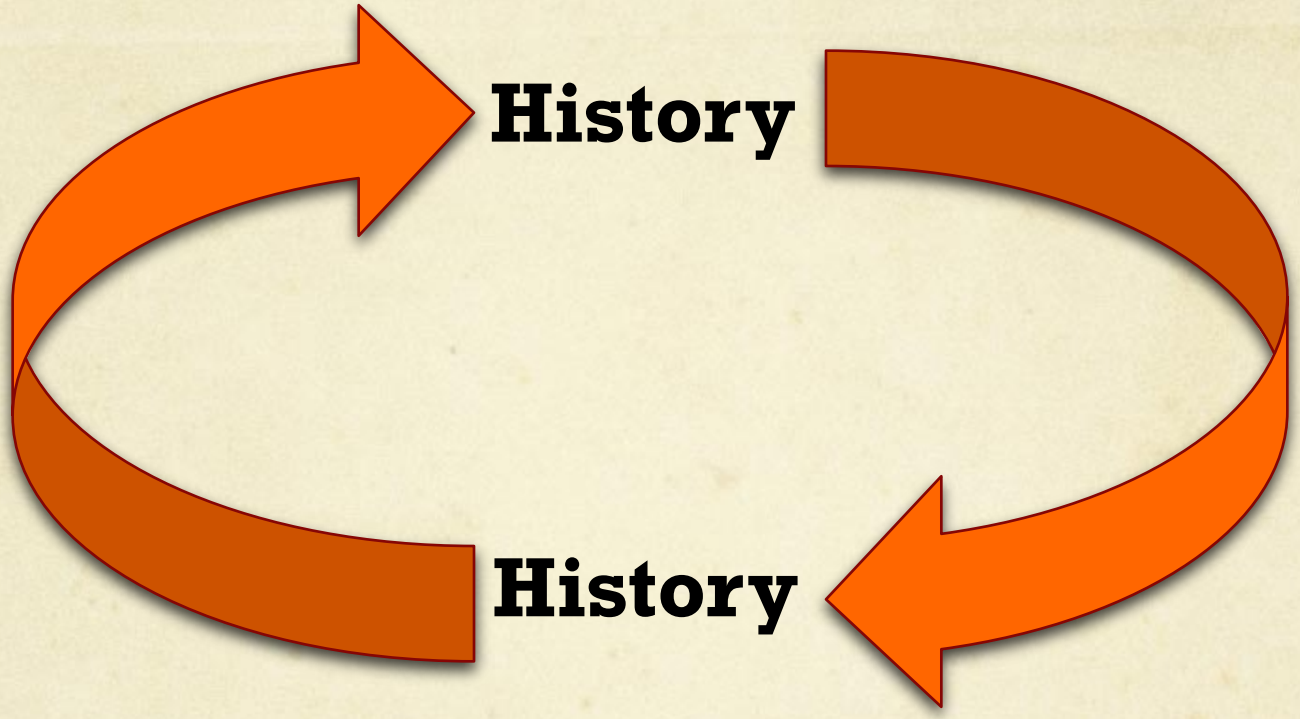
Services integration from anywhere on the planet to any device.

Why Am I Mentioning This?

Why Am I Mentioning This?

Services integration is important.
Asynchronicity is next.

A Primary Tenet of Computing



If History Repeats Itself, Is There No Future?

goo.gl/ebiksb

KAAZING

Reactive Programming, Streams and Events – The Alive Web

The Reactive Manifesto

Reactive programming is programming with asynchronous data streams.

Apache Kafka
A high-throughput distributed messaging system.

Spark

Amazon Kinesis

Clojure

Reactive-Extensions / RxJS

Watch 288

The Reactive Extensions for JavaScript http://reactivex.io

1,934 commits | 2 branches | 101 releases | 180 contributors

Branch: master RxJS / +

mattpodwysocki Merge branch 'master' of https://github.com/Reactive-Extensions/RxJS Latest commit e45382 7 hours ago

.nuget Updating NuGet packages

dist Closes Issue #974

RxJava: Reactive Extensions for the JVM

RxJava is a Java VM implementation of Reactive Extensions: a library for composing asynchronous and event-based programs by using observable sequences.

It extends the observer pattern to support sequences of data/events and adds operators that allow y uences together declaratively while abstracting away concerns about things like l synchronization, thread-safety and concurrent data structures.

node.js

Java 8

Functional Reactive Programming

goo.gl/ebiksb

Web Communication Models for Asynchronous Streams

goo.gl/ebiksb

KAAZING

Web Communication Protocols for Event-Driven World

HTTP/1.1 - 1997
RFC 2068

Great for caching, synchronous req/resp, XHR for client async polling (AJAX), Comet for push

HTTP/2 - 2015
RFC 7540

Binary, mux over TCP, header compression, server can push into client cache, AJAX/Comet, 30-50%+

SSE HTML5 - 2009
W3C

Standardization of Comet (push), uni-directional, uses HTTP

WebSocket - 2011
RFC 6455

Binary/Text, full-duplex, persistent connection, *TCP for the Web*

goo.gl/ebiksb

KAAZING

Web Communication Mechanisms for Event-Driven World

Web Notifications
2015 - W3C

Browser Notifications outside webpage, can use with Service Workers (and WS)

Push API
2015 - W3C

Scripted access to push data, use with Service Workers (and WS)

2007

User-defined HTTP callbacks (POST)

2009

Event-driven architecture, non-blocking I/O API, JS for server

goo.gl/ebiksb

KAAZING

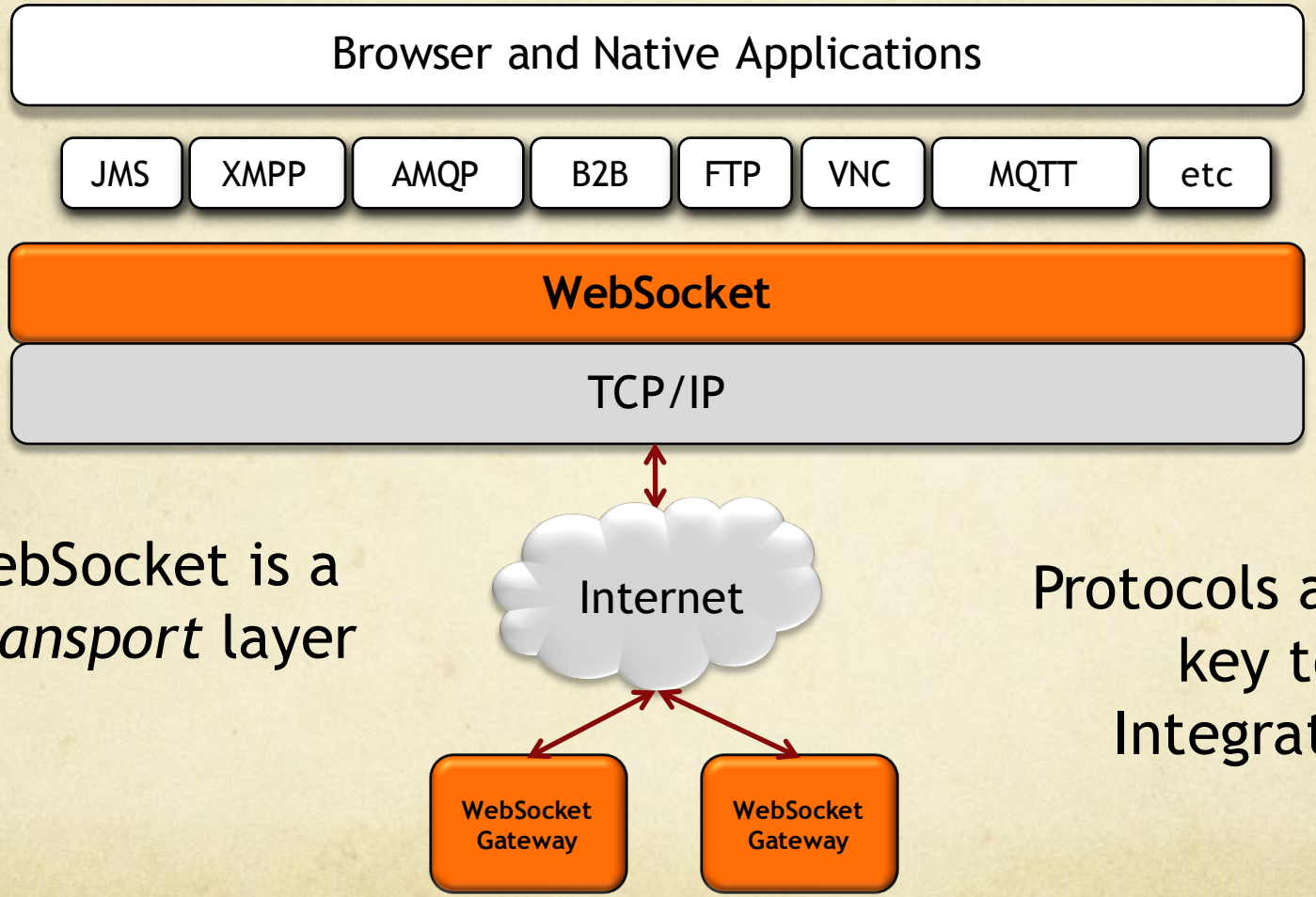
Do I Still Use WebSocket with HTTP/2?

- WebSocket is not a REST replacement
- WebSocket is complementary to HTTP (and REST)
- Simple Notifications can be easily done with HTTP
- WebSocket used for Full-duplex Persistent connection… a *TCP for the Web*
- Non-Browser use is where it gets interesting

WebSocket Projects

- Kaazing
- Node.js/socket.io/SockJS/engine.io
- ActiveMQ
- Tomcat
- Jetty
- Oracle Glassfish
- Java EE – **JSR 356**
- Play Framework – Reactive Apps
- Rabbit MQ
- JBoss
- IIS/ASP .NET 4.5
- PHP, Objective-C, Ruby, Python, C/C++, JVM-langs…
- Many more… (100+ implementations)

Protocol Layering is Possible



WebSocket is a *Transport* layer

Protocols are the key to Integration

goo.gl/ebiksb

KAAZING

What do Protocols give us?

Who handles retries?

How can we guarantee delivery?

How do we handle publish/subscribe semantics?

What do we do with slow consumers, last value cache, etc?

How do we handle market data?

What if the client is not currently active?

How do I handle entitlements? ACL?

What about partial data?

WoT Now?

The Web of Things

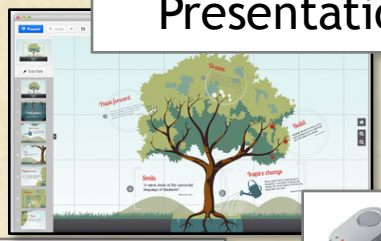
Internet of Things (IoT)

+

Heterogeneity + Scale + Usability

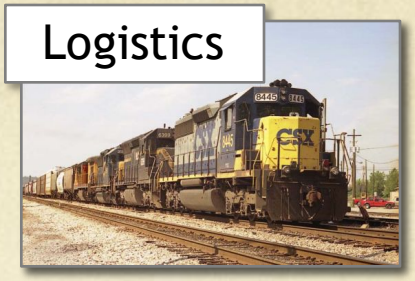
The World is Naturally Event-based ("real-time")

Communication

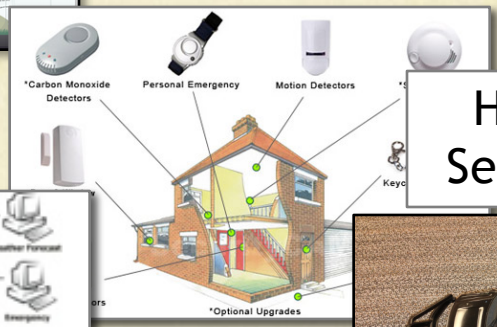


Presentation

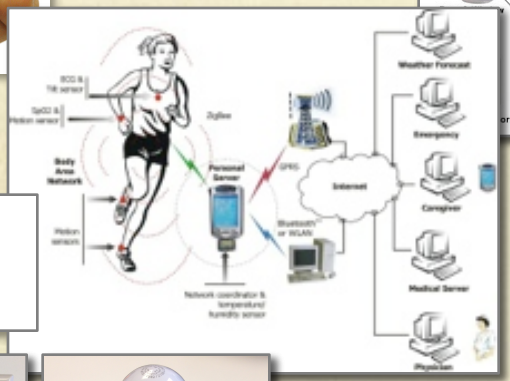
Music



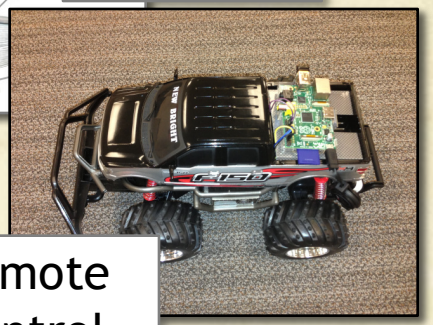
Logistics



Home Security



Health Monitoring



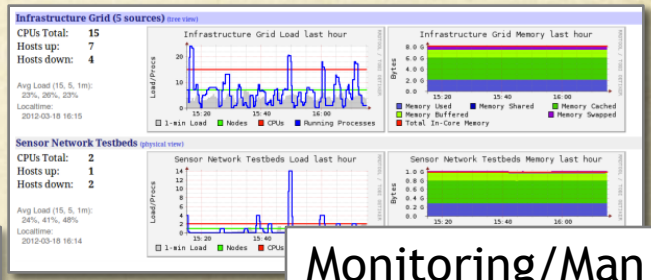
Remote control

Big Data

Risk Management

Intelligent Appliances

Local Transportation

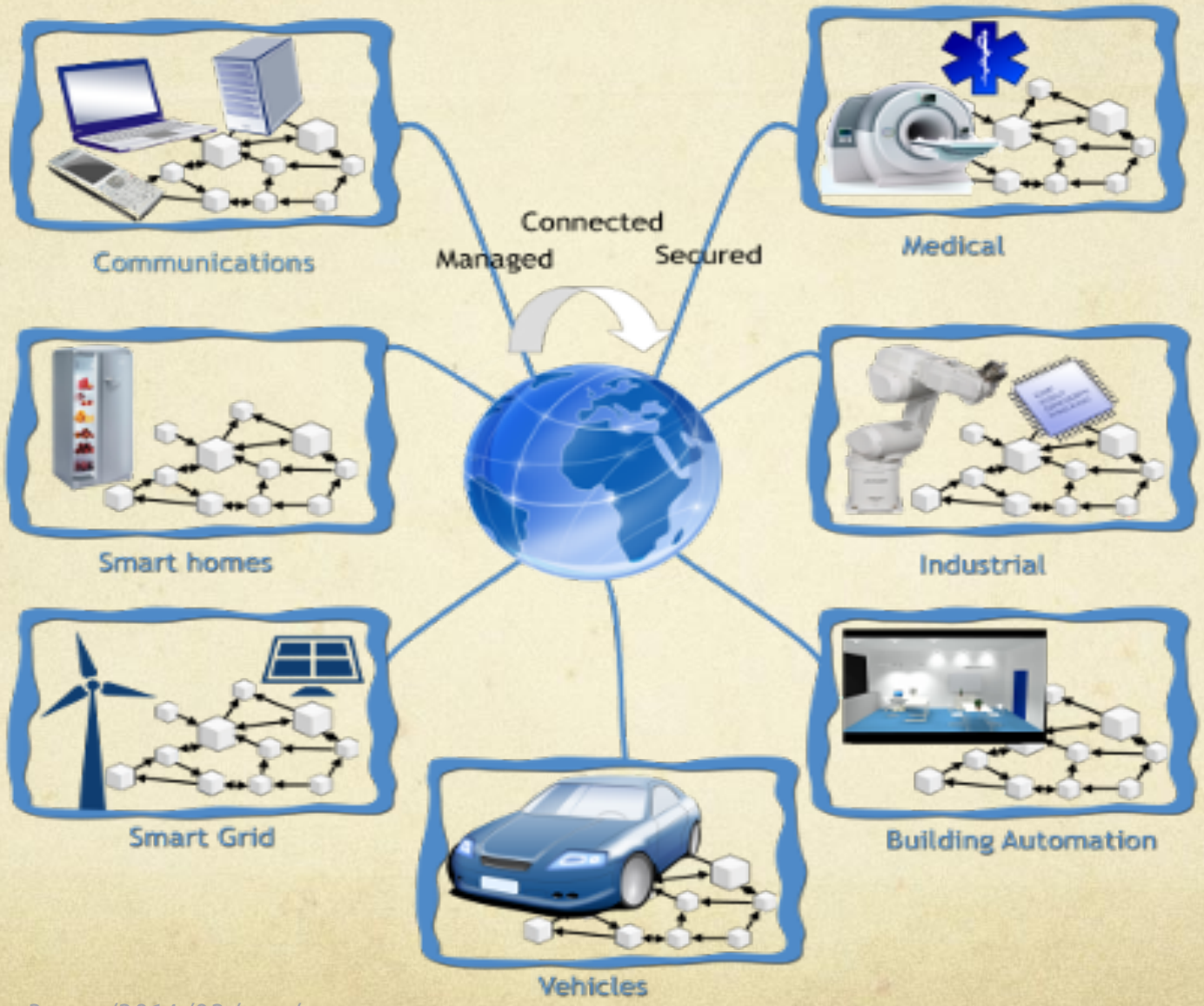


Monitoring/Management

goo.gl/ebiksb

KAAZING

Web of Things – Its All About SERVICES!



http://www.w3.org/2014/02/wot/

goo.gl/ebiksb

KAAZING

WoT does this have to do with the Web?

IoT/IIoT – Connectivity isn't Sufficient

- No formal API standards
- Many protocol standards – interoperability low
- No common, wide-reaching frameworks
- No composition possibilities
- Difficult to leverage economies of scale
- Barrier to entry is high for millions of app developers

- Also… we're in a cloud, mobile, web api world

Here's Where the Web Comes In

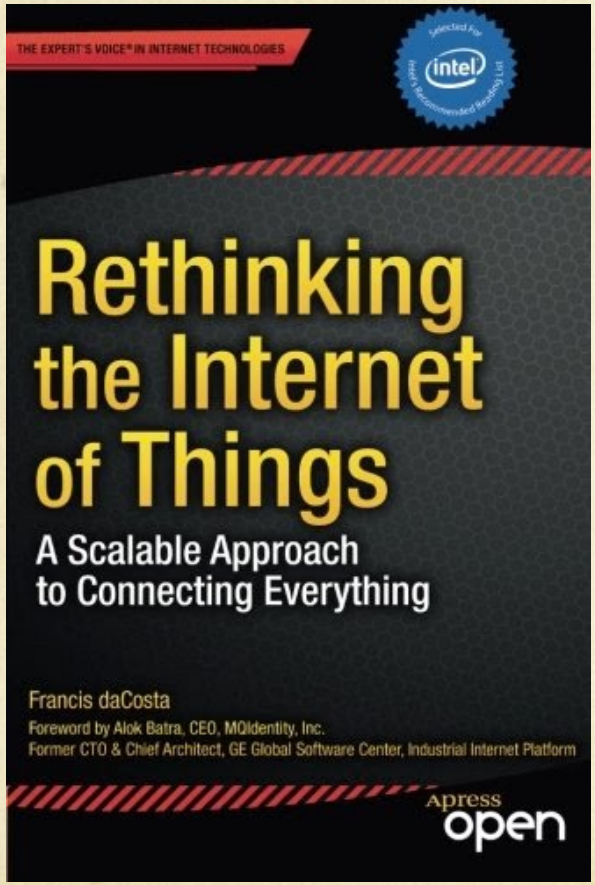
- IoT – Internet of Things
  - Embedded computing endowed with Internet connectivity
- WoT – Web of Things
  - Application and Services layer over IoT

- Apply the benefits of the Web to IoT
- WoT is a uniform interface to access IoT functionality
- Provides the abstraction for control/monitoring (sensors/actuators)
- Accelerates innovation
- Deployment, development, interoperability, economy of scale…

But Is HTTP the Right Choice?

- Disadvantages of HTTP Request/Response
- Lack of resiliency and robustness
- Enterprise events retrieved by resource intensive polling techniques
  - Much bandwidth is wasted
  - Information can be delayed
- Composite services brittle and lack transactionality
- Enterprises learned advantages of ESB 10+ years ago
- See failures of CORBA, Sun RPC, etc.
- Clumsy AJAX/Comet workarounds to simulate real-time

The Message is the Medium

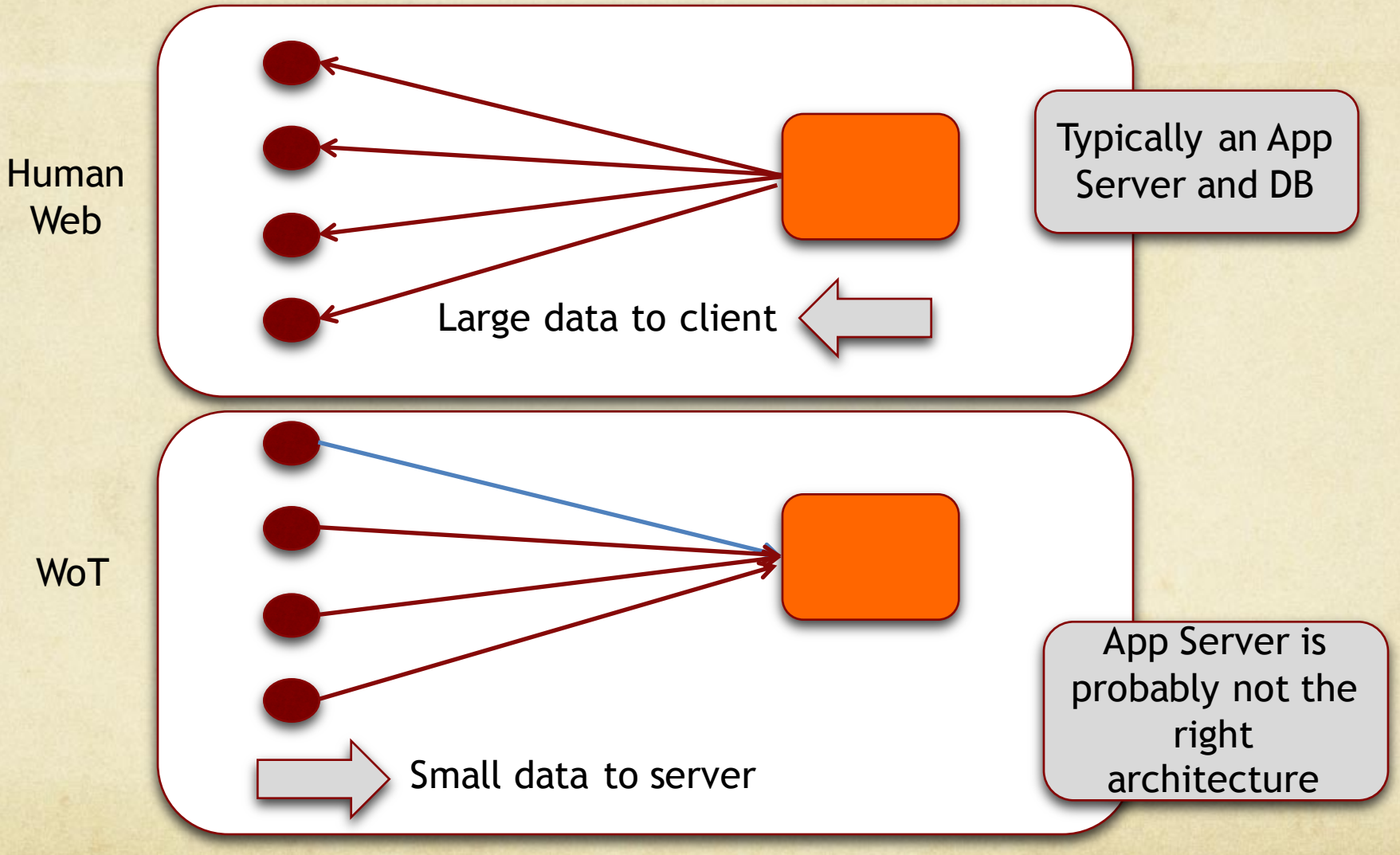


"…terse, self-classified messages, networking overhead isolated to a specialized tier of devices, and ***publish/subscribe*** relationships are the only way to fully distill the power of the coming Internet of Things" – Francis daCosta

goo.gl/ebiksb

KAAZING

Data Flow – Human Web vs WoT

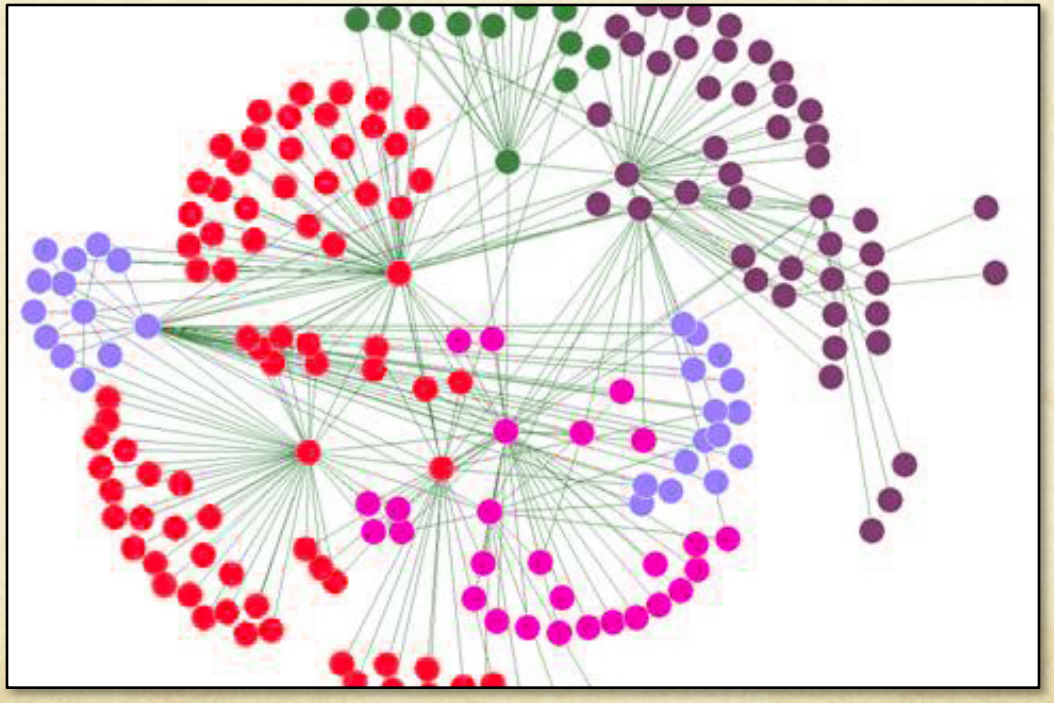


Do human-readable protocols make sense for non-humans?

goo.gl/ebiksb

KAAZING

Microservices



goo.gl/ebiksb

KAAZING

Why are we talking about Microservices?

- It's an SOA (*lightweight SOA*)
- It's SOA without WS-*, SOAP, etc, crap
- Older technique now useful with modern infrastructure
- An App is a Collection of Services
- Nothing really "micro" about Microservices
- If you need more than two pizzas to feed the team with the largest service, its not small enough

Monoliths vs Microservices

Monoliths

- Long builds, complex internals, scale issues
- Scale by replicating entire monolith on multiple servers
- Hard to modify
- Not necessarily bad – depends on team

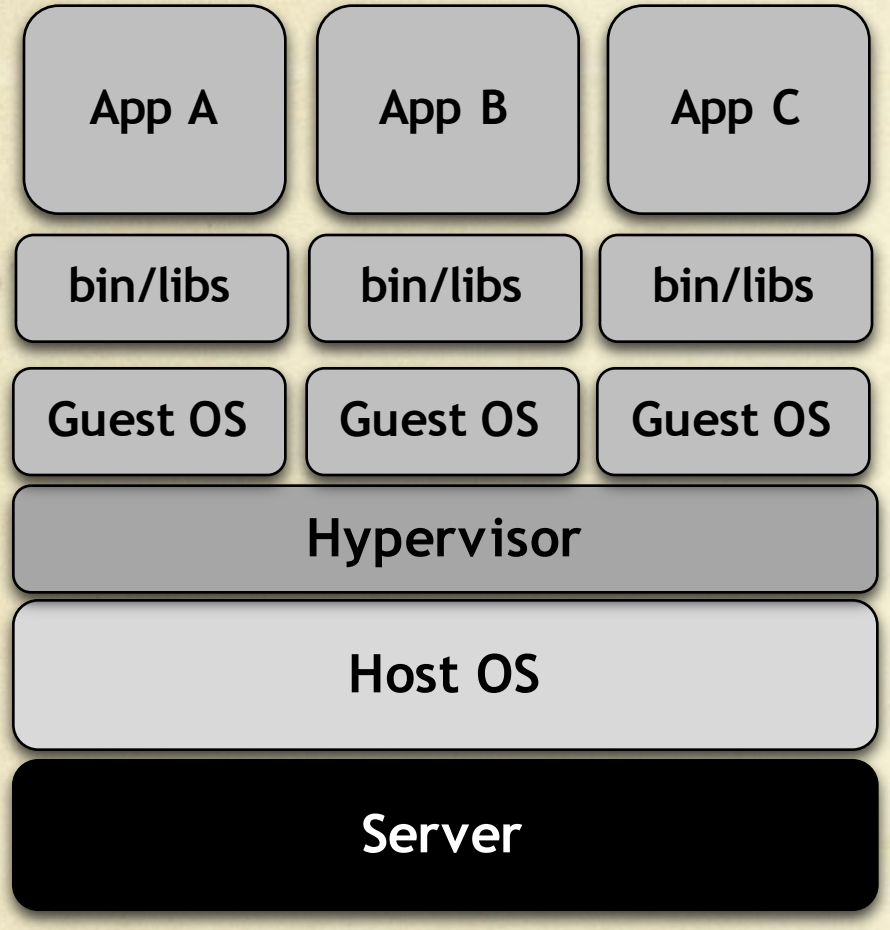
Monoliths vs Microservices

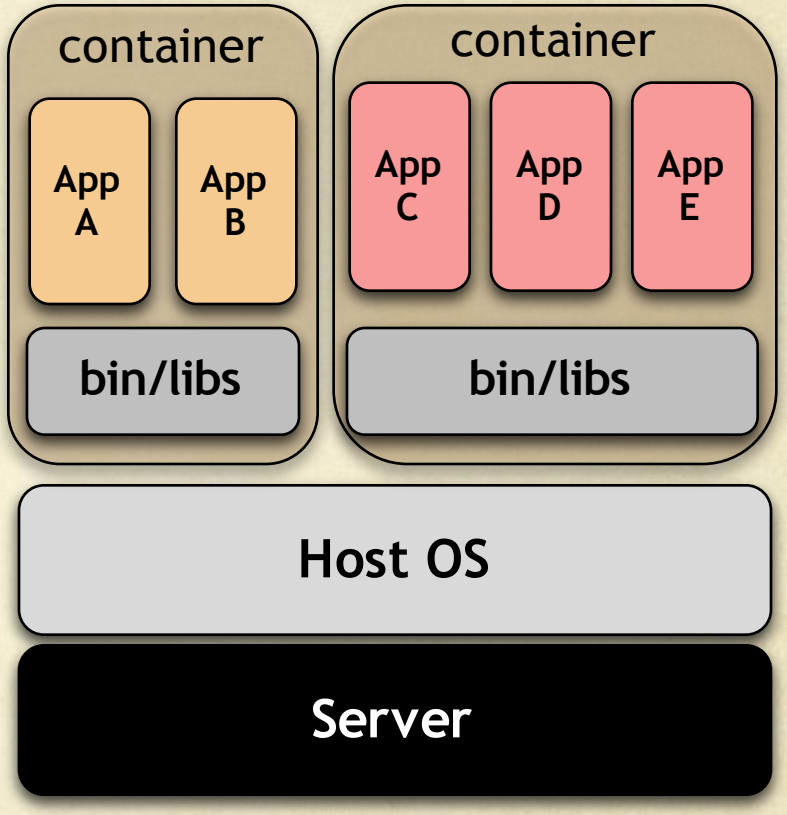
Microservices

- Small services – more agile
- Scale by replicating services
- Independent distributed services
- The Unix way
  - `% cat myfile | tr "A-Z" "a-z" | tr -cs 'a-z' '\n' | sort | uniq`
- Requires more management
- Still early

But we've had this idea for a while…
Let's take a step back

Containers vs Virtual Machines (VM)





goo.gl/ebiksb

KAAZING

Clouds and Microservices – the bottom line

- More services per an OS
- Greater Services Mobility – dev and ops
- Easier application patching
- Faster provisioning
  - 10 min for VM, 10 sec (or less) for microservice
- Container internals visible to help maximize optimization
- Avoids cloud framework lock
- Intercloud portability
- Allows services to be located most appropriate part of architecture
- Allows policies to be applied per container

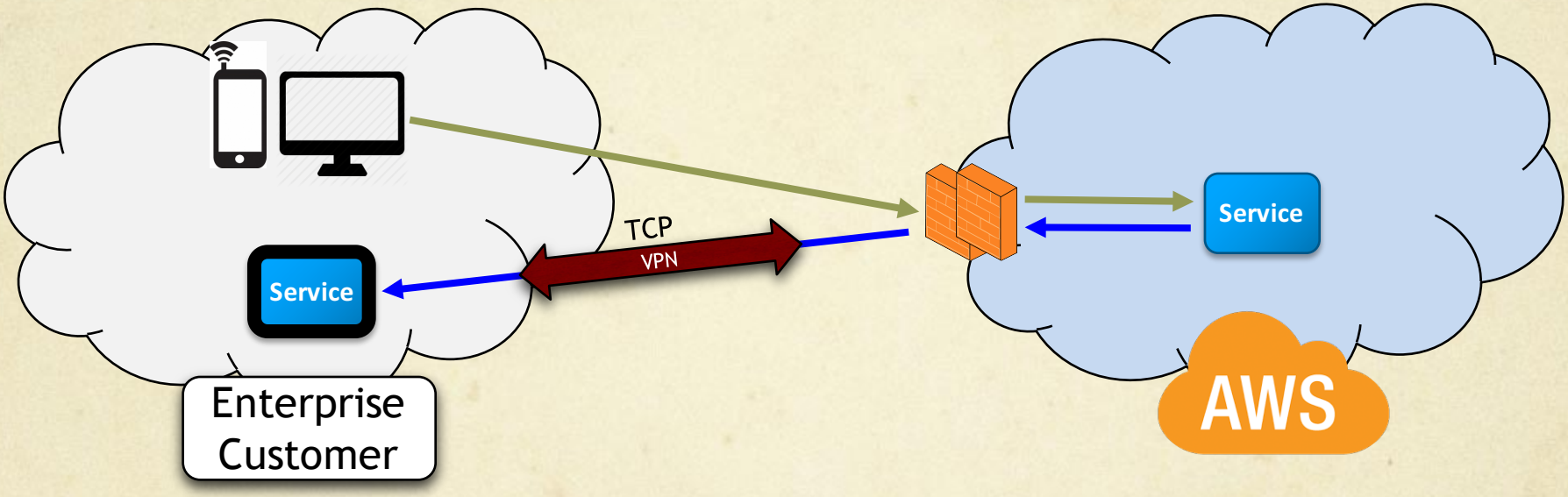
Clouds and Microservices – the bottom line

- The Microservices Synchronicity Penalty
- Many ecommerce sites use 150-200 microservices for personalization. Amazon.com
- Many are REST-based… ie, synchronous (wait for a reply). And many are chained, so the penalty is additive.
- Significant resources are needed for high levels of scalability (S = G/C)

Cloud Connectivity

WebSocket for Hybrid Cloud Connectivity

Cloud services frequently require on-premises access



- Access must be on-demand, secure and real-time
- Requires lengthy VPN installation process, open ports or worse

goo.gl/ebiksb

KAAZING

Demos

WIN A COPY!

goo.gl/ebiksb

KAAZING

WIN A COPY!

1. Introduction to HTML5 WebSocket
2. The WebSocket API
3. The WebSocket Protocol
4. Building Instant Messaging and Chat over WebSocket with XMPP
5. Using Messaging over WebSocket with STOMP
6. VNC with the Remote Frame Buffer Protocol
7. WebSocket Security
8. Deployment Considerations

goo.gl/ebiksb

Stretching Web Communication to Its Limits

goo.gl/ebiksb

KAAZING

Stretching Web Communication to Its Limits

goo.gl/ebiksb

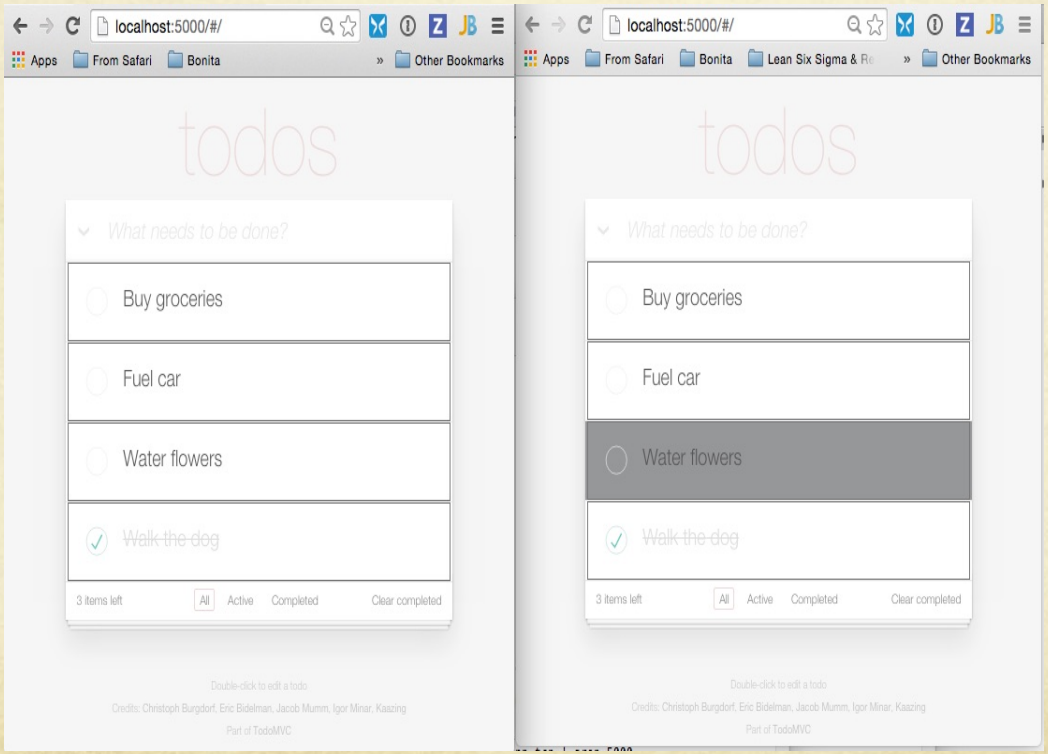
KAAZING

TodoMVC – Angular with Kaazing

TodoMVC

Helping you **select** an MV* framework

Download | View on GitHub | Blog



goo.gl/ebiksb

KAAZING

Stretching Web Communication to Its Limits

Browser - Server

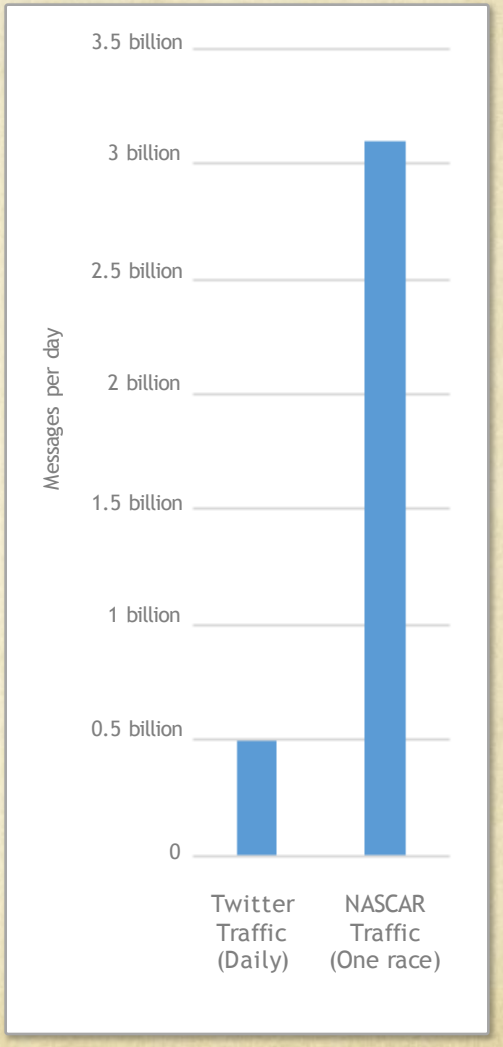
Native (mobile, desktop) - Server

goo.gl/ebiksb

KAAZING

Over a Billion Messages an Hour

In a single 3½ hour race,
Kaazing broadcasts
over 6 times as many messages
as Twitter does in an entire day



goo.gl/ebiksb

KAAZING

Stretching Web Communication to Its Limits

Browser - Server

Native (mobile, desktop) - Server

IoT/Embedded - Server

goo.gl/ebiksb

KAAZING

Stretching Web Communication to Its Limits

Browser - Server

Native (mobile, desktop) - Server

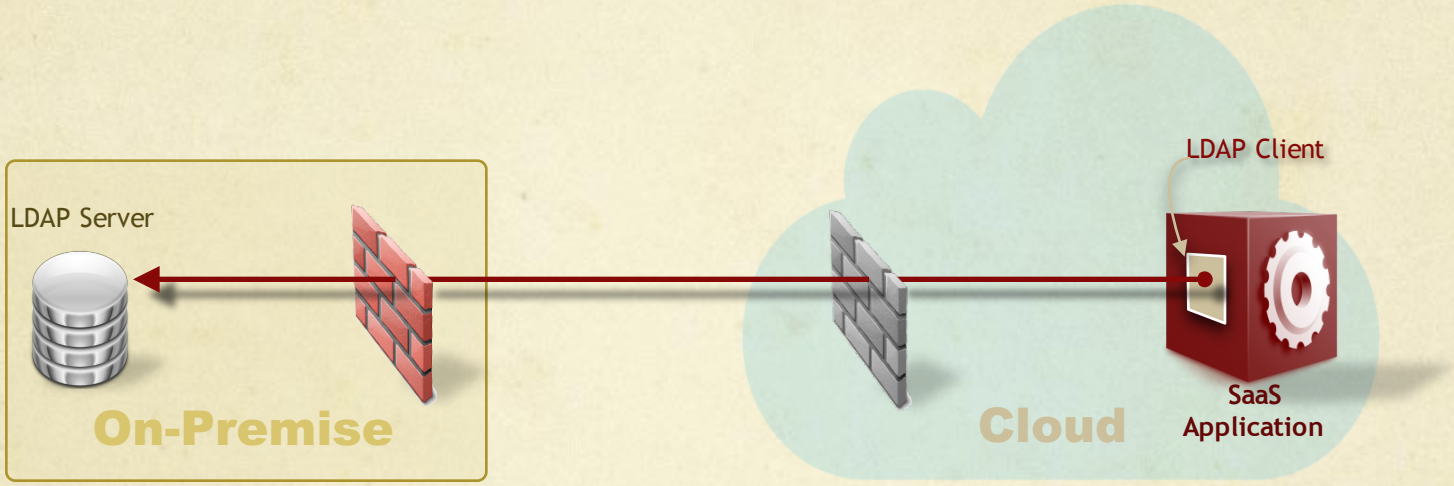
IoT/Embedded - Server

Server - Server

goo.gl/ebiksb

KAAZING

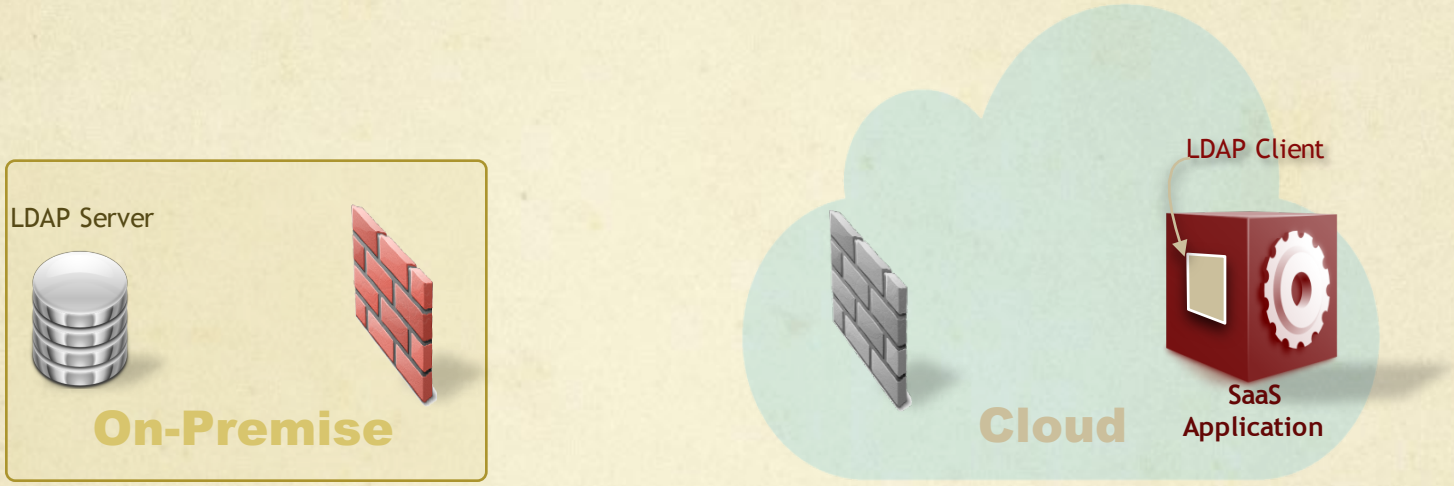
A2A for B2B. No VPN Needed.



goo.gl/ebiksb

KAAZING

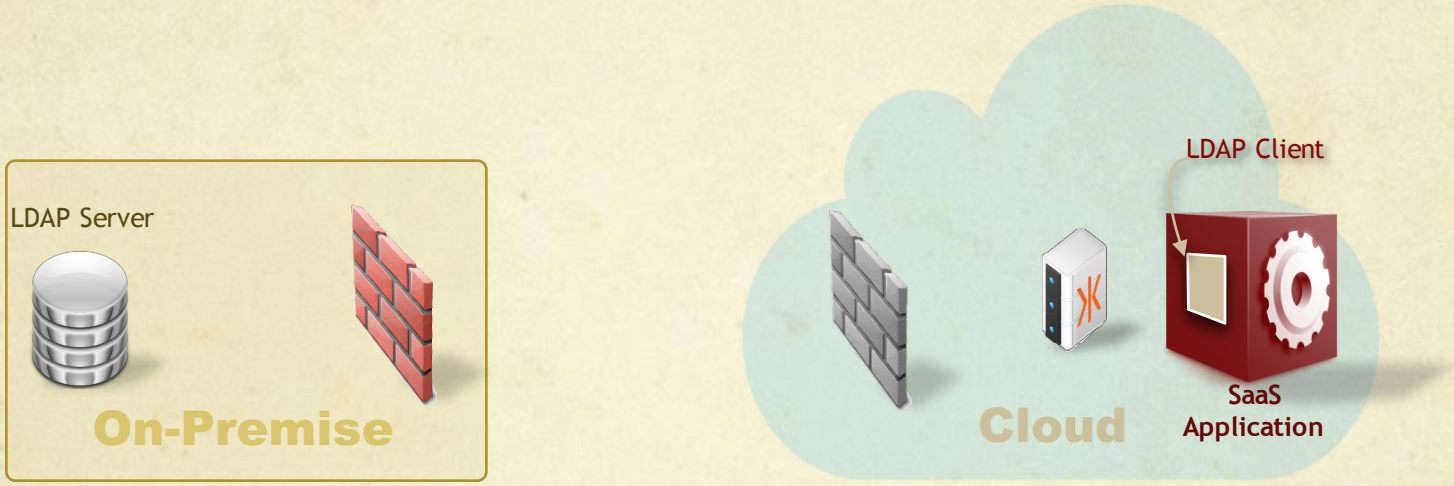
A2A for B2B. No VPN Needed.



goo.gl/ebiksb

KAAZING

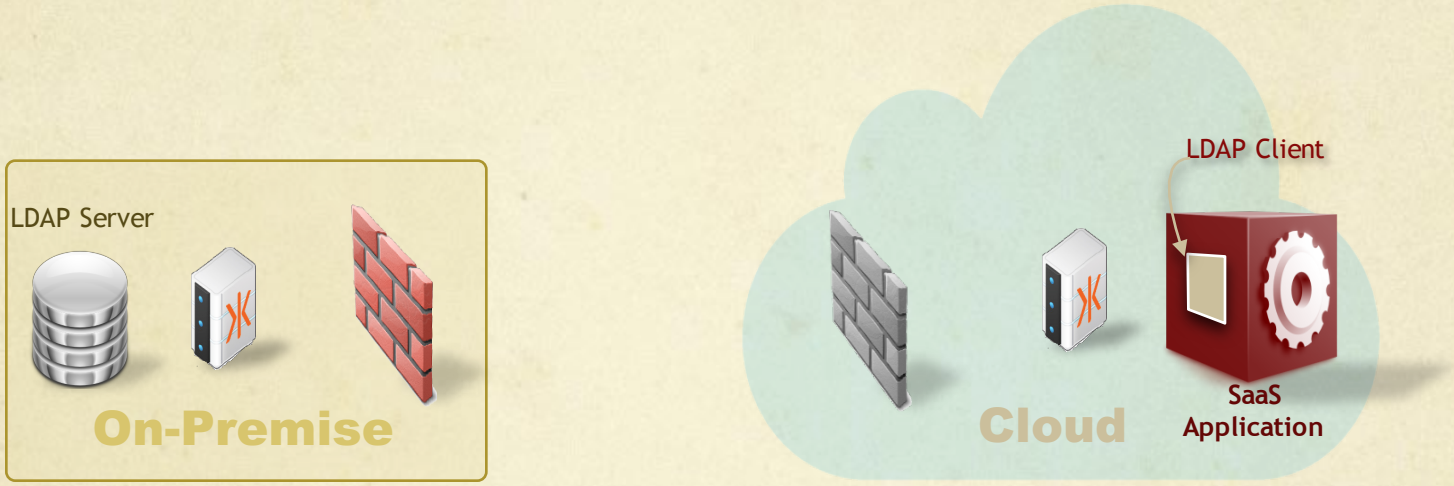
A2A for B2B. No VPN Needed.



goo.gl/ebiksb

KAAZING

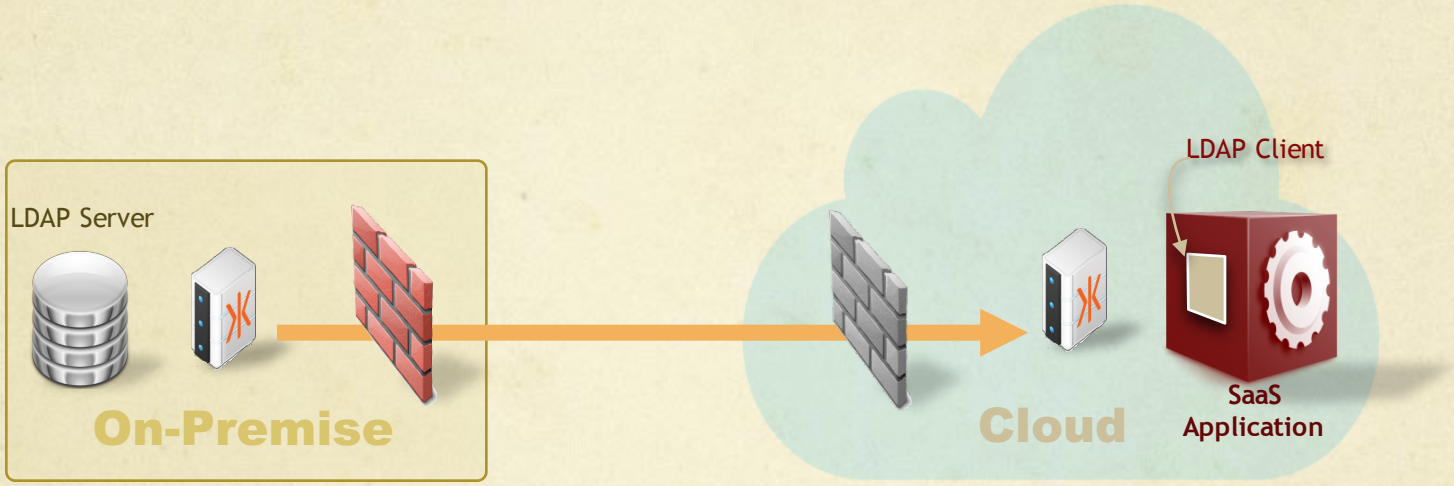
A2A for B2B. No VPN Needed.



goo.gl/ebiksb

KAAZING

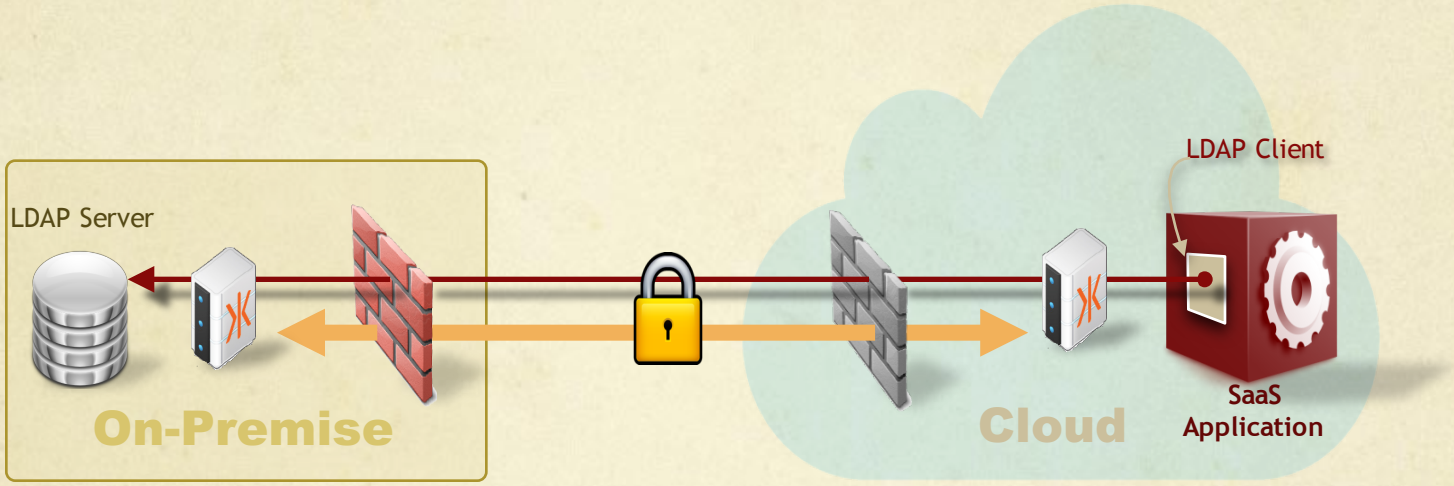
A2A for B2B. No VPN Needed.



goo.gl/ebiksb

KAAZING

A2A for B2B. No VPN Needed.



goo.gl/ebiksb

KAAZING

WIN A COPY!

1. Introduction to HTML5 WebSocket
2. The WebSocket API
3. The WebSocket Protocol
4. Building Instant Messaging and Chat over WebSocket with XMPP
5. Using Messaging over WebSocket with STOMP
6. VNC with the Remote Frame Buffer Protocol
7. WebSocket Security
8. Deployment Considerations

goo.gl/ebiksb

Raffle time

```
$> jot -r 1 2 99
```

goo.gl/ebiksb

KAAZING

Thank You!

@frankgreco

@pmoskovi

goo.gl/ebiksb

KAAZING