RABBITMQ ON OPENVMS

Brett Cameron & John Apps
Taxi Drivers
May, 2010

AGENDA

– **AMQP**

– Erlang

– RabbitMQ

– Open Source Software Projects on OpenVMS

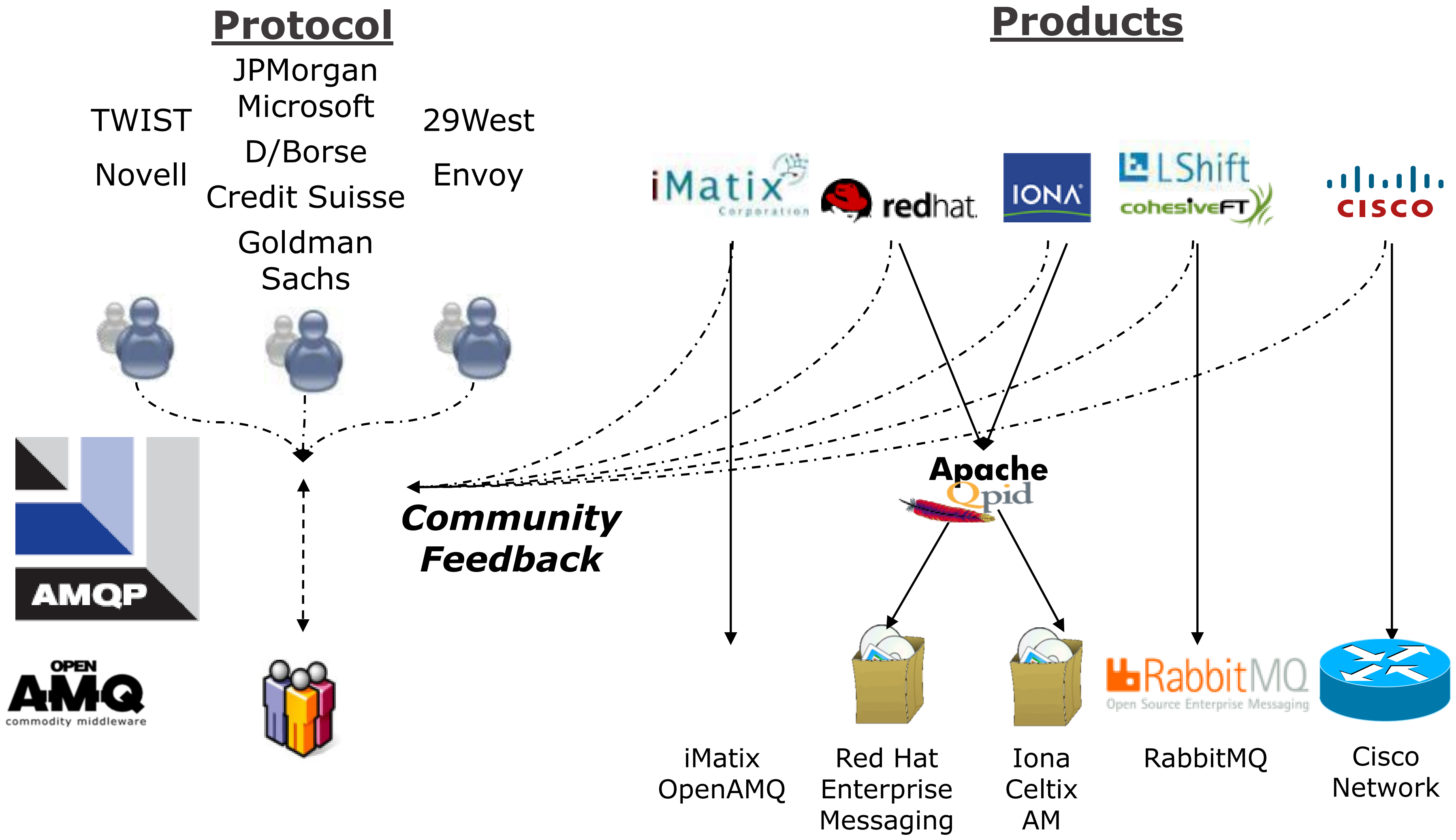
AMQP

- AMQP (Advanced Message Queuing Protocol) is an open standard application layer protocol for message oriented middleware
- The defining features of AMQP are:
  - Message orientation and Queuing
  - Routing (including point-to-point and publish-and-subscribe)
  - Reliability and Security
- AMQP mandates behaviour of messaging provider and client
  - Implementations from different vendors truly interoperable
  - Previous attempts to standardise middleware focussed at API level and failed
  - AMQP therefore defines a wire-level protocol
    - A wire-level protocol describes the format of data sent across the network as a stream of octets
    - Any tool that can create and interpret messages conforming to the defined wire-level protocol can interoperate with any other compliant tool, irrespective of implementation language

AMQP

- AMQP was originally designed to provide a vendor-neutral protocol for managing the flow of messages across an enterprise's business systems
- AMQP was developed from mid-2004 to mid-2006 by JPMorgan Chase and iMatix
  - iMatix also developed the original AMQP implementation (OpenAMQ)
- JPMorgan Chase and iMatix documented the protocol and assigned it to a working group that included:
  - Red Hat, Cisco Systems, TWIST, IONA, and iMatix
- As of August 2009, the working group consists of:
  - Barclays Bank, Cisco Systems, Credit Suisse, Deutsche Börse Systems, Envoy Technologies, Goldman Sachs, IONA, iMatix, JPMorgan Chase, Microsoft, Novell, Rabbit Technologies, Red Hat, Solace Systems, Tervela, TWIST, WSO2, 29West, …
- Although AMQP originated in the financial services industry, it has general applicability to a broad range of middleware problems

AMQP



Protocol

TWIST

Novell

JPMorgan
Microsoft
D/Borse
Credit Suisse
Goldman Sachs

29West

Envoy

Community Feedback

Products

iMatix OpenAMQ

Red Hat Enterprise Messaging

Iona Celtix AM

RabbitMQ

Cisco Network

AMQP
OPEN AMQ commodity middleware
iMatix Corporation
redhat
IONA
LShift
cohesiveFT
CISCO
Apache Qpid
RabbitMQ Open Source Enterprise Messaging

AMQP Working Group controls the standard

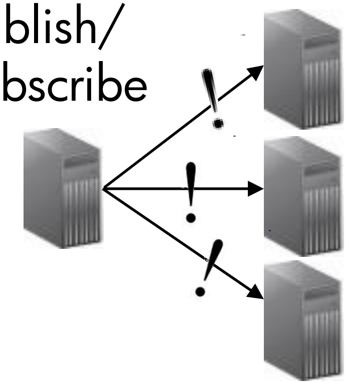
Diverse products implement the standard

AMQP

– Key features of AMQP include:
  - Queuing with strong delivery assurances
  - Event distribution with flexible routing
  - Large message capability (gigabytes)
  - Global addressing scheme (email-like)
  - Meets common requirements of mission-critical systems
  - Service oriented

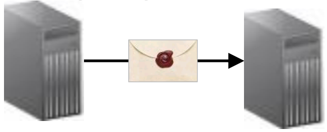
Any language, any model, any payload, any transport, any platform, reliable, interoperable, manageable, performant, scalable

Publish/ subscribe



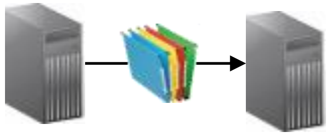
detect

Messaging



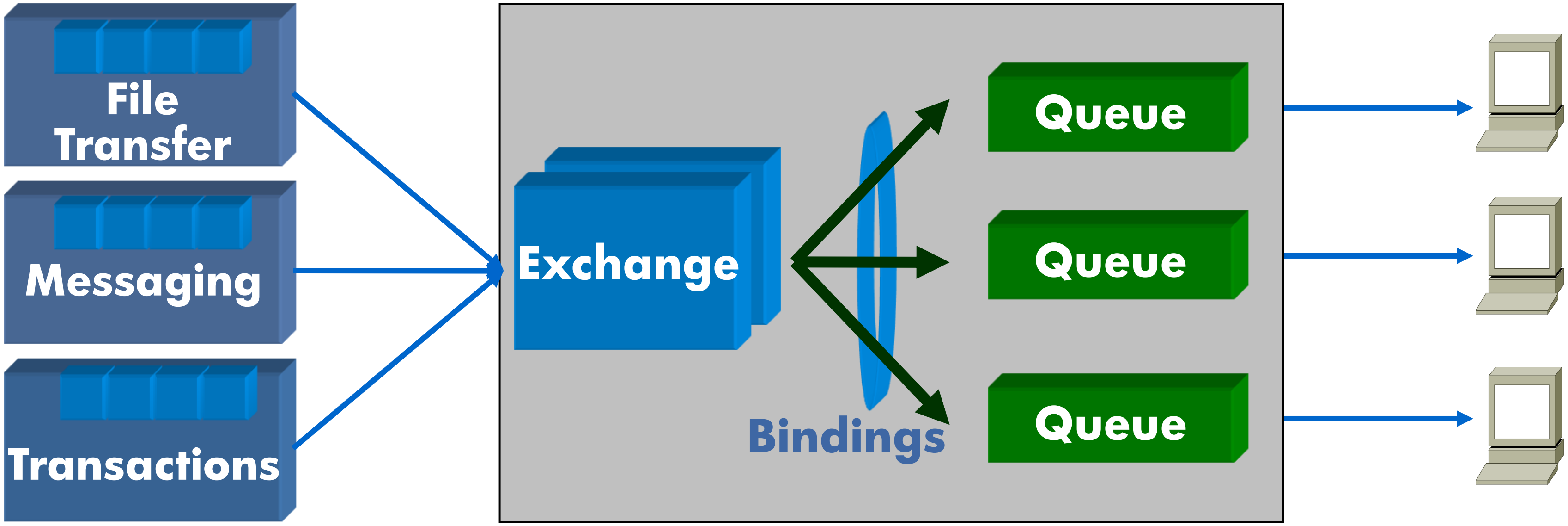
transact

File transfer



report

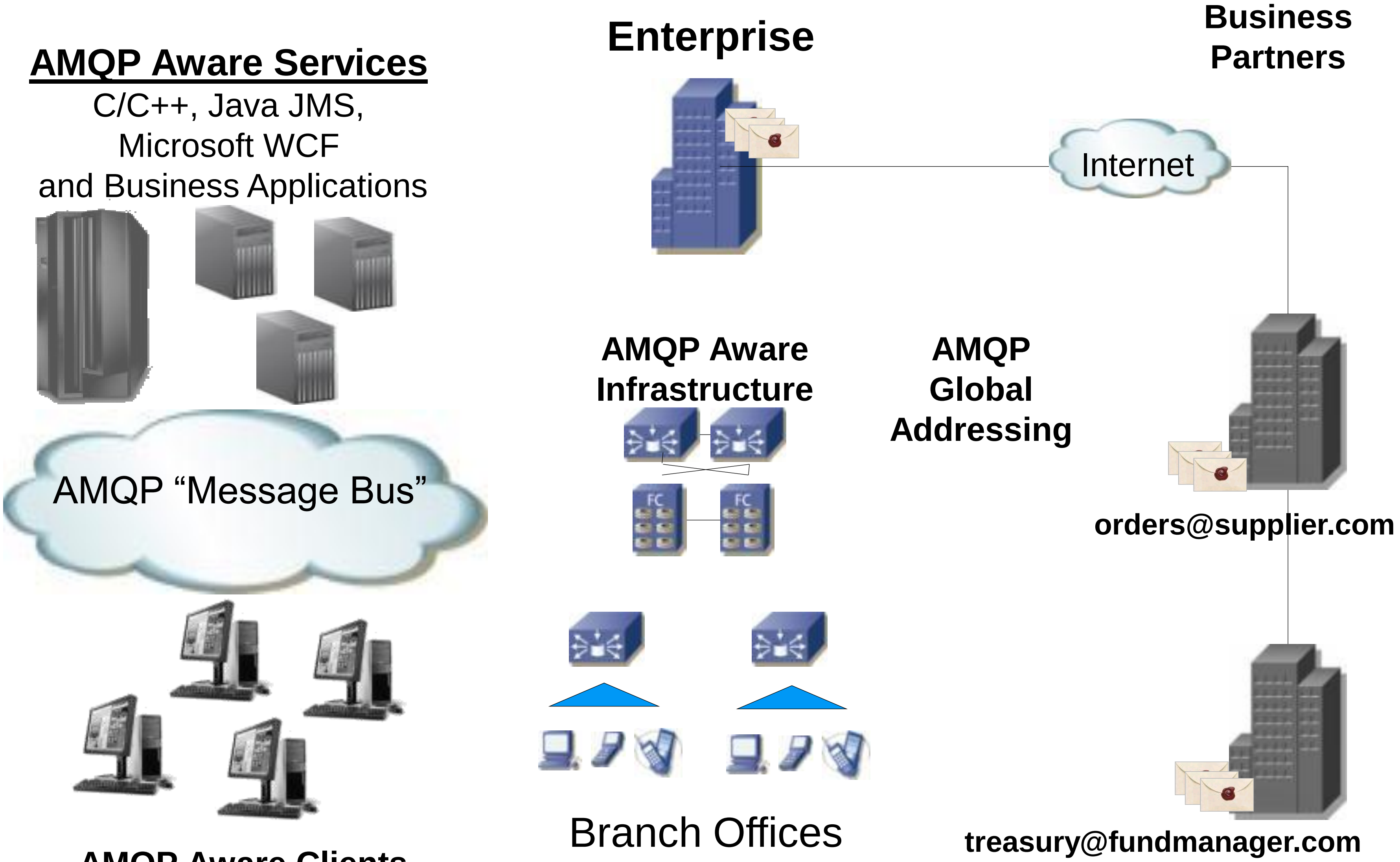
AMQP



- any language *(C, C++, Java, C#, Python, Javascript, Erlang, Lisp, Ruby, Tcl, PHP, …)*
- any model *(native, JMS, Mule, WCF, …)*
- any payload *(binary, XML, SOAP, JSON, …)*
- any transport *(TCP, SCTP, HTTP, …)*
- any platform *(desktop, router, mobile, EC2, …)*

- reliable
- interoperable
- manageable
- performant
- scalable

AMQP VISION



AMQP Aware Services

C/C++, Java JMS,
Microsoft WCF
and Business Applications

AMQP "Message Bus"

AMQP Aware Clients

Devices & workstations

Enterprise

AMQP Aware Infrastructure

Branch Offices

AMQP Global Addressing

Business Partners

Internet

orders@supplier.com

treasury@fundmanager.com

AGENDA

– AMQP

– **Erlang**

– RabbitMQ

– Open Source Software Projects on OpenVMS

ERLANG

- Erlang is a programming language designed at the Ericsson Computer Science Laboratory.
- Open-source Erlang is being released to help encourage the spread of Erlang outside Ericsson

- Article in Ericsson Review No. 1, 1997. Describes the Open Telecom Platform (OTP). OTP is written in Erlang.

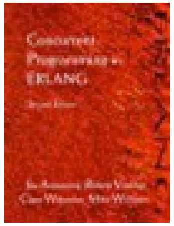
  "The open telecom platform (OTP) is a development system platform for building telecommunications applications, and a control system platform for running them. The platform, whose aim is to reduce time to market, enables designers to build – from standard, commercially available computer platforms – a highly- productive development environment that is based on the programming language Erlang."

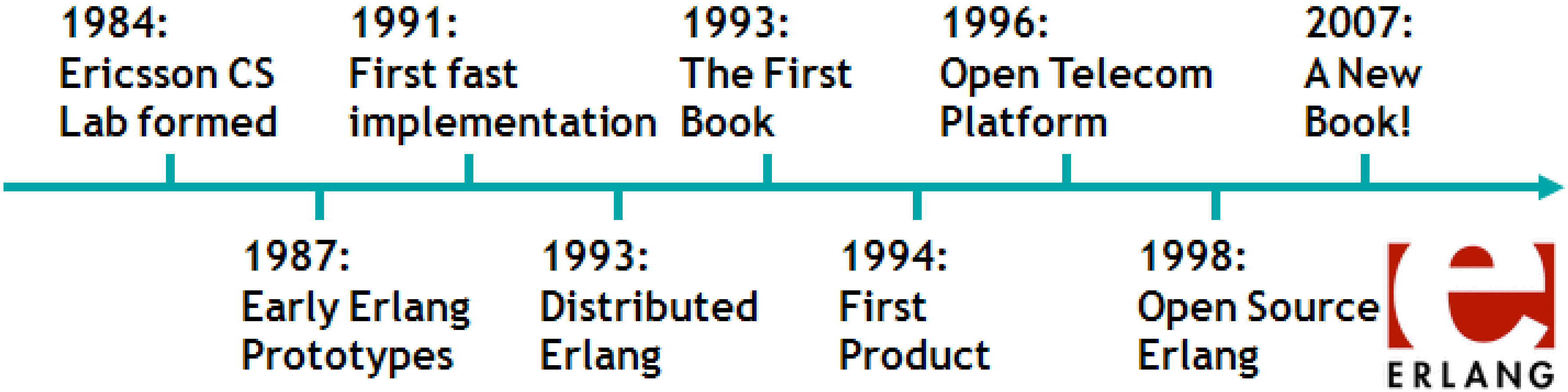
  "The OTP also permits application designers who program in C, C++, Java and other languages to take full advantage of sourced components. Moreover, the OTP allows designers to consider costs when matching computer platforms with requirements for processing power and component availability."

ERLANG

- In his search to find better ways to implement fault-tolerant applications, **Joe Armstrong** designed and implemented the first version of Erlang in 1986

No language was well suited for telecom systems development




1984: Ericsson CS Lab formed

1987: Early Erlang Prototypes

1991: First fast implementation

1993: Distributed Erlang

1993: The First Book

1994: First Product

1996: Open Telecom Platform

1998: Open Source Erlang

2007: A New Book!

ERLANG

- **Concurrency** - Erlang has extremely lightweight processes whose memory requirements can vary dynamically
- **Distribution** - Erlang is designed to be run in a distributed environment. An Erlang virtual machine is called an Erlang node
- **Robustness** - Erlang has various error detection primitives which can be used to structure fault-tolerant systems
- **Soft real-time** - Erlang supports programming "soft" real-time systems, which require response times in the order of milliseconds
- **Hot code upgrade** - Erlang allows program code to be changed in a running system
- **External interfaces** - Erlang processes communicate with the outside world using the same message passing mechanism as used between Erlang processes
- **Compiler** generates byte code which is executed and optimised at runtime, i.e., a virtual machine
- **No-SQL databases** – one of the most popular languages for implementing no-SQL databases, e.g., CouchDB, mnesia

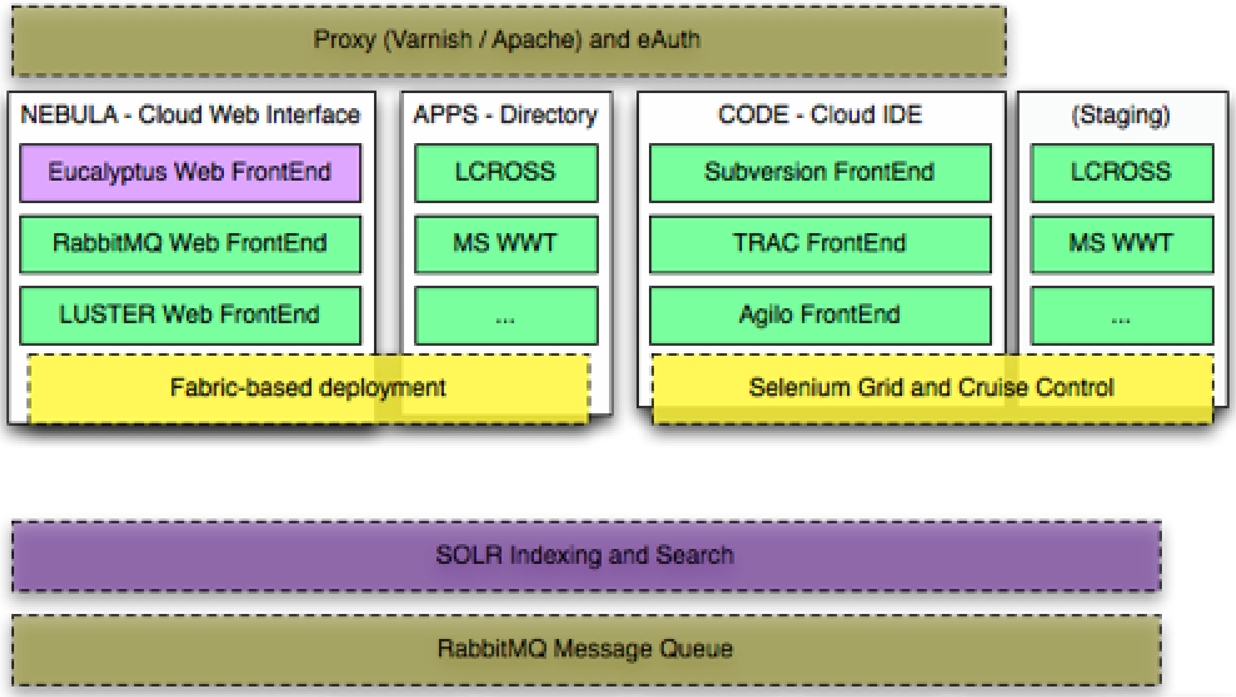
AGENDA

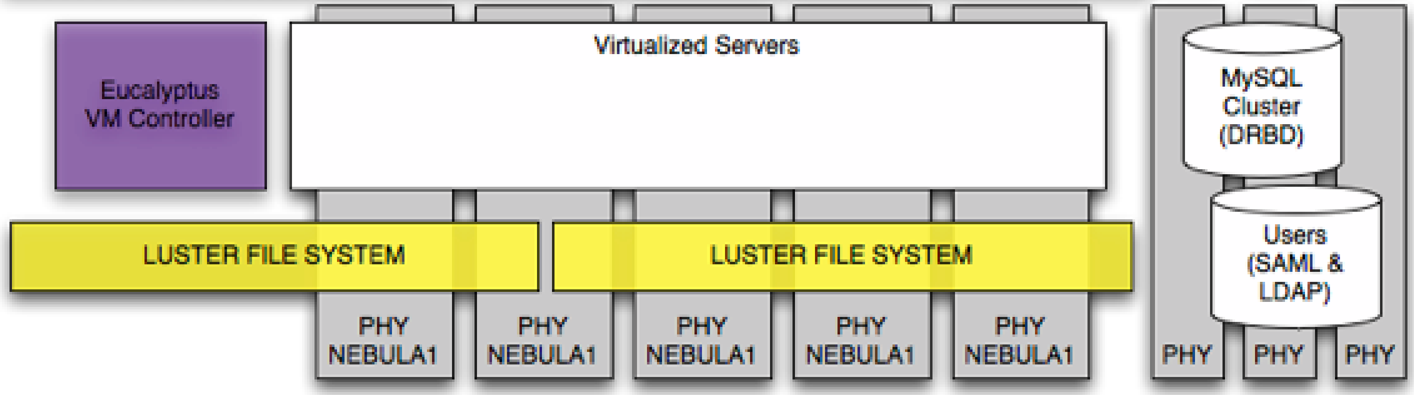
- – AMQP
- – Erlang
- – **RabbitMQ**
- – Open Source Software Projects on OpenVMS

RABBITMQ

- Founded in Feb 2007, purchased by SpringSource in April, 2010
- Similar to IBM MQ and Tibco RV – no “lock in”
- The goal is to lower total messaging costs by 75%
- Is a member of the AMQP technical committee
- Implements AMQP 0.8 and 0.9.1
- Is written in Erlang
- Maintains clients for
  - C, Java and .NET

NASA Cloud ("Nebula") also used by US Government "apps.gov"





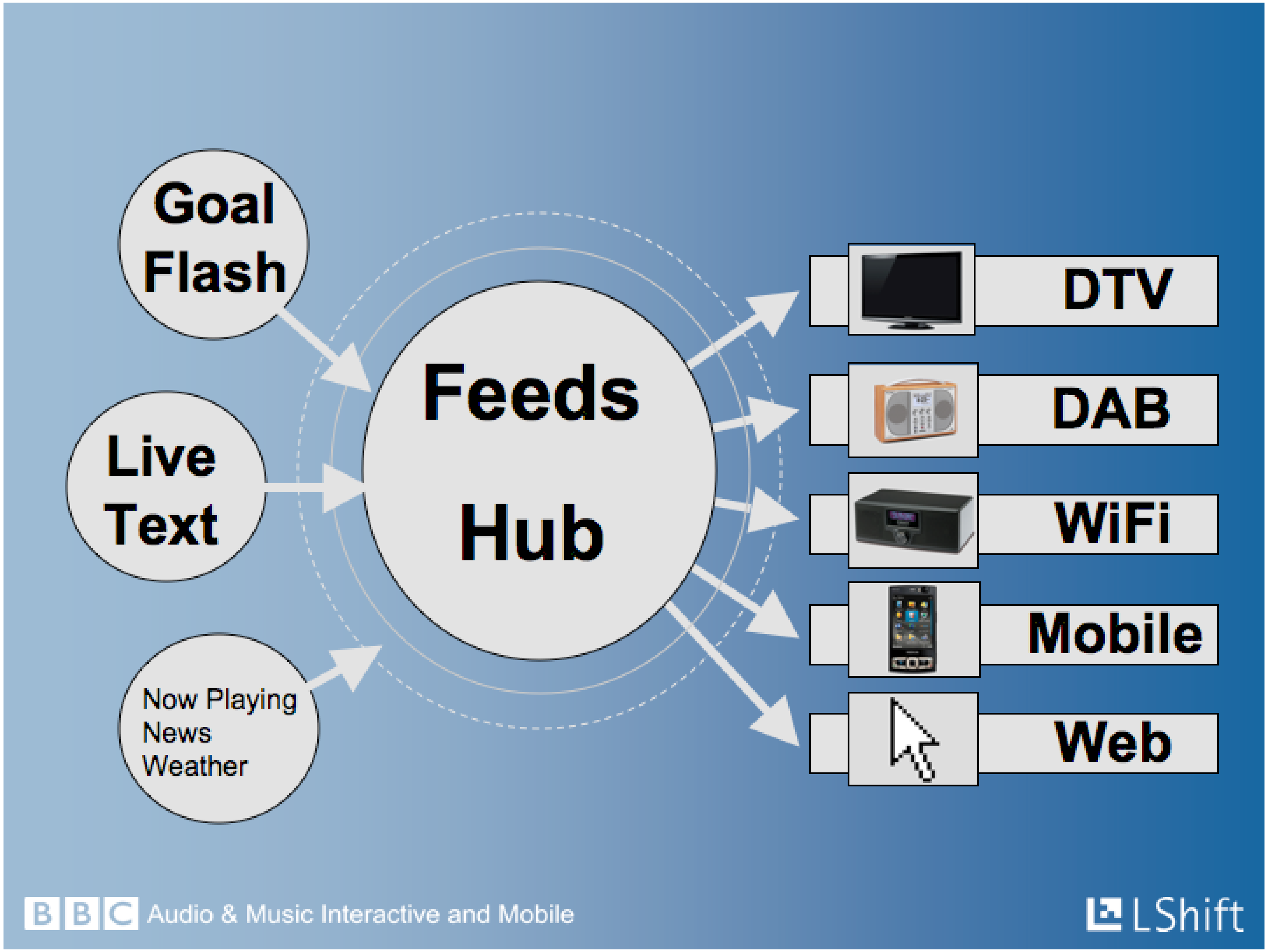
🟩 Python

🟪 JAVA

🟫 Other

RabbitMQ
Open Source Enterprise Messaging

BBC - REAL TIME CONTENT MANAGEMENT



RABBITMQ

Queues

- Can be persistent
- Can be (automatically) deleted when empty
- Can be (automatically) deleted when server stops

RABBITMQ

Messages

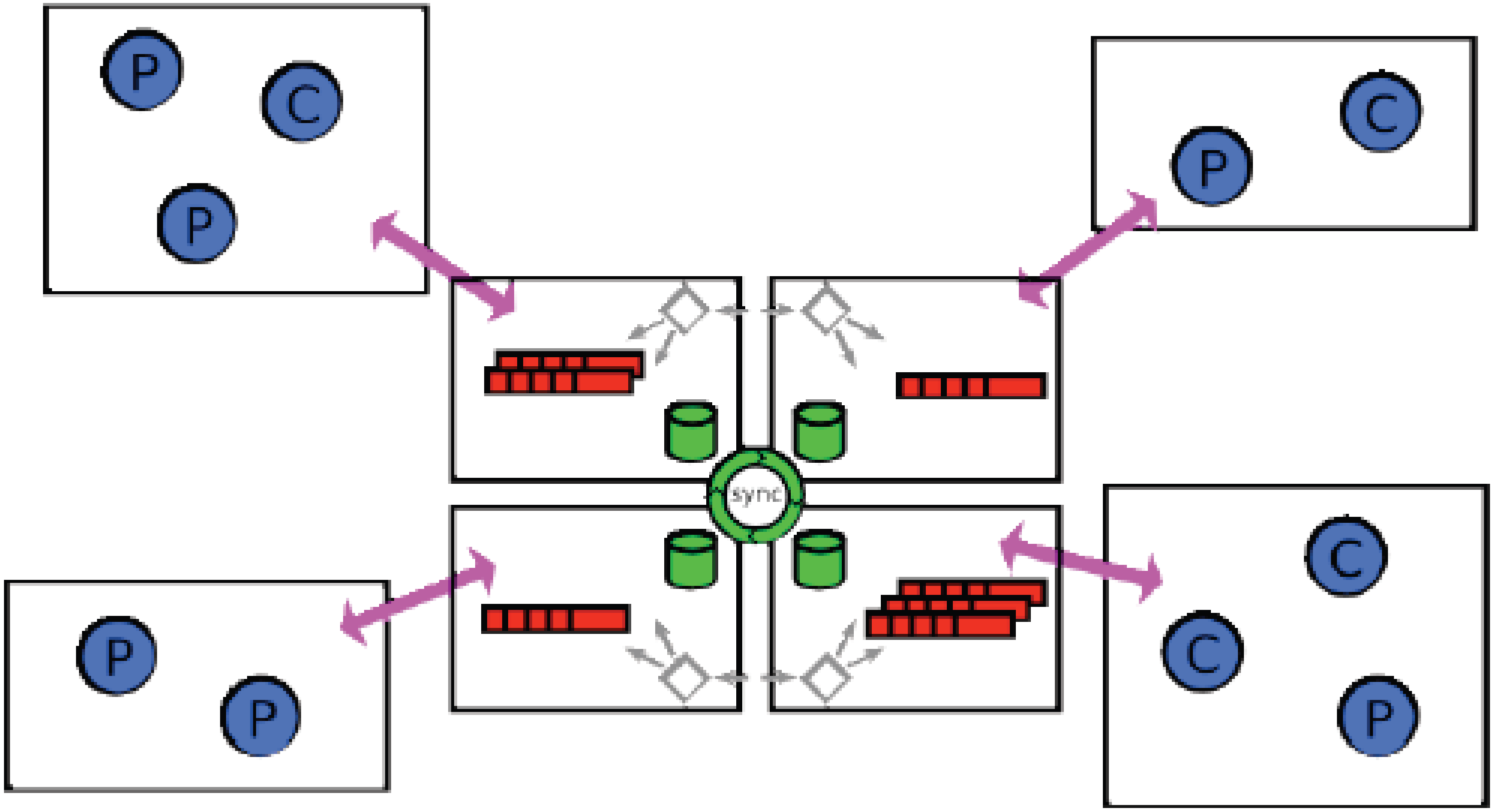
- Can be persistent
- Can be in memory only
- Might be persisted if not enough memory
- Flow control when consumer cannot keep up with producer
- …

RABBITMQ CLUSTERING

– Multiple nodes share routing information

– Queues are distributed

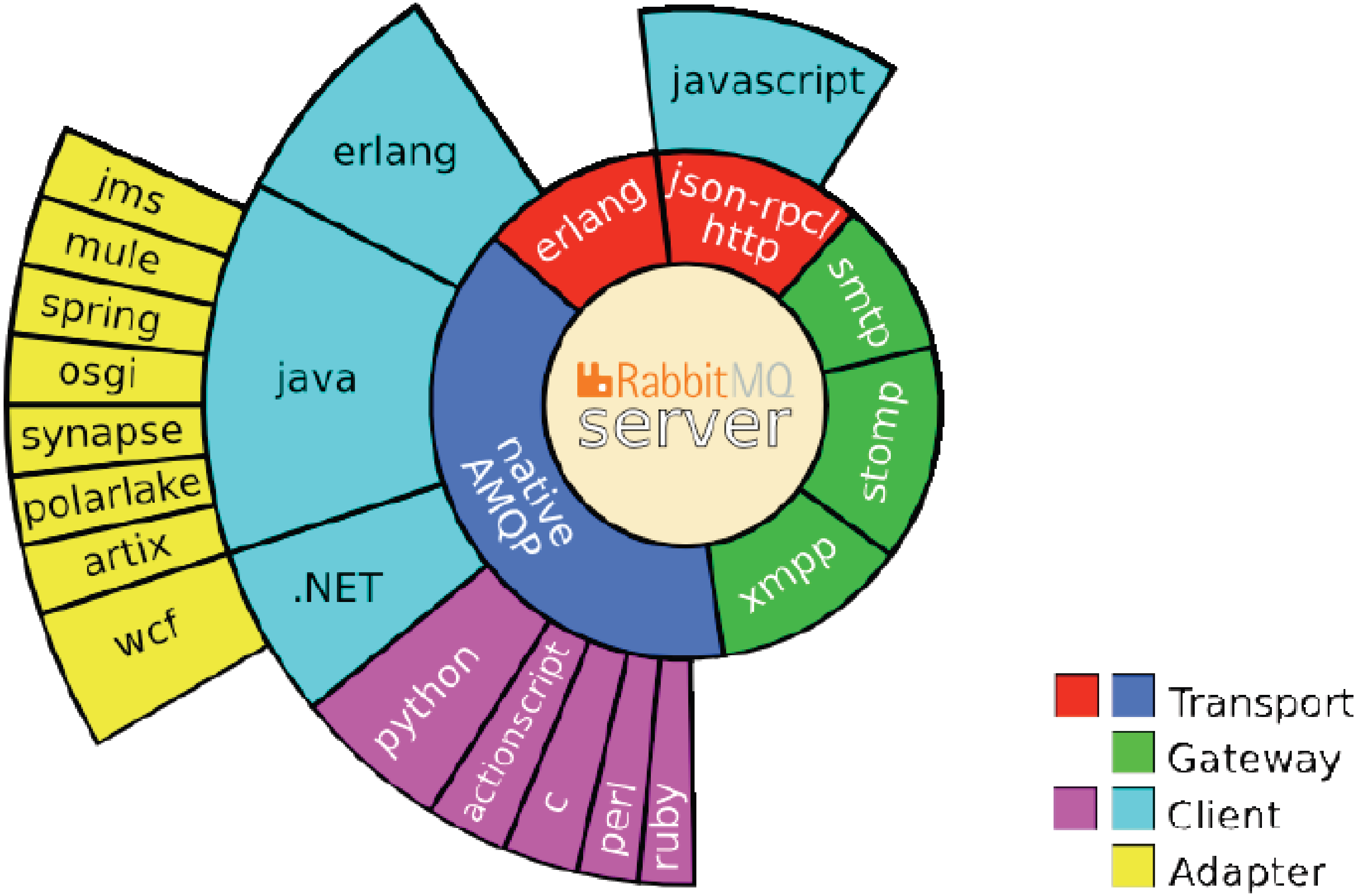
– Messages are routed efficiently between nodes



NOTE: Clustering comes 'out of the box' with Erlang

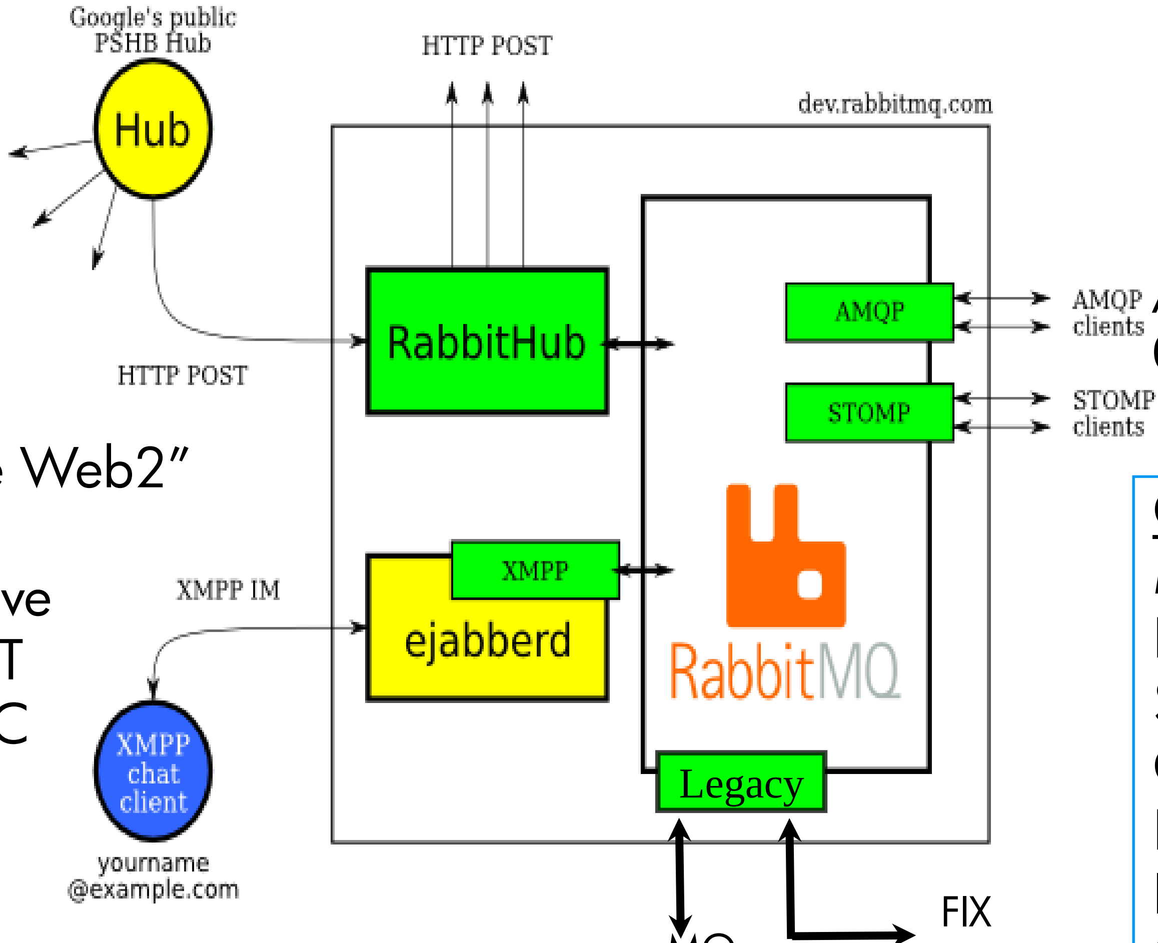
RABBITMQ

– Client Libraries



RABBIT 2: VIRTUAL MESSAGING BROKER JOINING LEGACY TO CLOUD

“Real time Web2”
XMPP IM
XMPP Wave
HTTP/ReST
JSON/RPC



AMQP
0-91 and 1.0

MQ
RDBMS
MSMQ
RTR?
Oracle Message Queue?

FIX
29west

Current R&D
Management
Rabbit Apps (plug-ins)
Shared Service
Cross Protocol
Remote Serviceability
Identity, Security
Storage and HA

RABBITMQ

- Implements AMQP 0.80 and 0.9.1
- RabbitMQ home page
- Runs on
  - Amazon EC2, Linux, UNIX, Windows, MAC, JavaScript, **OpenVMS**, …
- Maintains
  - C, Java and .NET interfaces (with WCF)
- The community implements and provides support for
  - Python, Ruby, PHP, Perl, Scala, Groovy, OSGi, Spring, COBOL (by HP), …
- Has interfaces/bridges for
  - HTTP (JSON-RPC), XMPP, JMS, memcached, PG_AMQP (AMQP extension for PostgreSQL), …
  - ***How about JCC LogMiner with AMQP extension?***
- RabbitMQ on OpenVMS

RABBITMQ - STATE OF PLAY ON OPENVMS

- Erlang ported to OpenVMS Integrity
  - Non-SMP version done; working on SMP version
  - Port not yet complete
    - Performance enhancements to be done
    - Not production quality
    - Lots of testing left to do
- Runs current release of RabbitMQ 1.7.2, including the mnesia DB
  - Using Java clients for testing ‘out of the box’
  - Interoperability testing between clients on other platforms and OpenVMS
- Ported libRabbitMQ client ‘C’ libraries
  - OpenVMS AMQP API for high-level languages (COBOL, FORTRAN) almost done
  - Working on gateways to enable integration with other messaging and TP products, e.g., ACMS, RTR, …
    - Hope to demonstrate the OpenVMS Advanced Technical Boot Camp in September, 2010 (if we can get there)

AGENDA

– AMQP

– Erlang

– RabbitMQ

– **Open Source Software Projects on OpenVMS**

OPEN SOURCE SOFTWARE PROJECTS ON OPENVMS

RELEASED

- gSOAP – Web Services implementation in C
- Libwww – W3C Protocol Lib
- OpenAMQ – AMQP implementation
- 0MQ – very lightweight messaging (10μ Latency, >8 million messages/second)
- FastCGI – CGI without sub-processes

TESTING

- Erlang – programming language and environment
- RabbitMQ – AMQP implementation (written in Erlang)
- Yaws – Yet Another Web Server (written in Erlang)
- LUA – lightweight scripting language
- Mongoose – Lightweight Web Server by Google

IN PROGRESS

- Twitter API – talk to Twitter
- Memcached – distributed caching
- Gearman – distributed load management
- Libevent – async notification library
- Tokyo Cabinet – High-performance key/value pair DB
- Google Protocol Buffers – extensible mechanism for serializing structured data

Thank you!