



**Open Source
Integration Options
and Ideas for
OpenVMS**



Abstract

www.vmssoftware.com

This talk will examine several Open Source products and open standard protocols that can be used on OpenVMS to integrate new and existing OpenVMS-based applications with applications on other platforms and with cloud-based services. Protocols such as AMQP and MQTT will be considered, along with various Open Source integration technologies that are available on OpenVMS, including the Mosquitto MQTT broker, Gearman, and Redis, along with Java-based solutions such as ActiveMQ, Tomcat, and WSIT. Technologies that can be used to implement and interact with SOAP-based and RESTful web services will also be considered. Examples of how these various technologies may be used will be presented and their applicability to particular problem domains will be discussed.

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

Introduction

www.vmssoftware.com

- Integration with other systems is a common requirement
 - Often perceived as difficult
 - Often perceived as expensive
- Traditionally dominant solutions are typically very proprietary
 - Frequently too expensive for everyday use
 - Invariably do not interoperate
- MQSeries, DEC/BEA/Oracle MessageQ, TIBCO, ...

Introduction

www.vmssoftware.com

- Such issues with proprietary solutions have resulted in numerous home-grown developments
 - Custom middleware solutions
 - Custom adaptors
- The net result for a large enterprise is middleware hell
 - Hundred's of applications, thousand's of links
 - Every other connection is different
 - Massive waste of effort
 - Costly to implement
 - Costly and difficult to maintain
- Open source solutions and open standards present a viable alternative...

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

RabbitMQ

www.vmssoftware.com



Im in yr serverz,
queueing yr messagez

RabbitMQ

www.vmssoftware.com

- A popular and powerful open source message broker
 - The leading implementation of AMQP
 - Provides a robust and flexible messaging platform designed to interoperate with other messaging systems
 - See <http://www.rabbitmq.com>
- A multi-protocol broker engine implemented in 100% Erlang/OTP for exceptional scalability and fault-tolerance

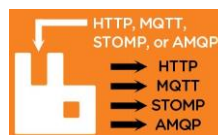


RabbitMQ

www.vmssoftware.com

Comprises the following components:

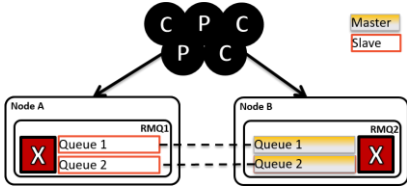
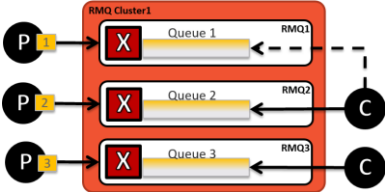
- The RabbitMQ broker
 - Natively supports AMQP 0.9.1
 - Supports AMQP 1.0 via a plugin
- Adapters for HTTP, STOMP, MQTT, and several other protocols
- Client libraries for Erlang, Java, .NET, and C/C++
- Clients for numerous other languages available from other vendors and/or the open source community
- Python, Ruby, PHP, Clojure, Node.js, Go, Rust, ...
 - Just about any language you can think of (even COBOL)
- Large assortment of useful plugins/extensions



RabbitMQ

- Scalability through clustering
- High-availability
- Management and monitoring
- Security
- Extensibility (via plugins)

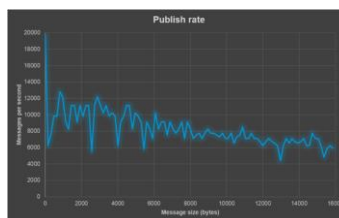
www.vmssoftware.com



RabbitMQ and OpenVMS

www.vmssoftware.com

- Need Erlang on OpenVMS in order to run RabbitMQ
- Have a 32-bit build of Erlang 19.1
 - Supports SMP
 - Can run RabbitMQ 3.6.x
 - Performance is "satisfactory"
 - Can achieve message rates (publish and consume) of up to ~19000 messages per second on an old dual 1.6GHz CPU rx4640 server using small messages
 - Largely limited by performance of Erlang/OTP TCP/IP driver code on OpenVMS
- Not sufficiently stable for production use
- Working on it 😊



RabbitMQ and OpenVMS

www.vmssoftware.com

- libRabbitMQ ported to OpenVMS
- Provides an API that can be used by OpenVMS-based software applications to exchange data via AMQP
 - Allows OpenVMS applications to publish and consume messages to and from RabbitMQ
- Based on C API originally developed by the RabbitMQ team
- Can be used with any OpenVMS 3GL (C/C++, Pascal, COBOL, FORTRAN, ..)
- Have also created tools using to simplify development of RabbitMQ clients
 - Generic consumer (loads and calls user-written functions in a shareable images)
 - Scripting utility (useful for testing)
 - Enhancements to WSIT to generate code for the RabbitMQ Java client
- Can also use Ruby and Python clients on OpenVMS

RabbitMQ and OpenVMS

www.vmssoftware.com

- Enterprise integration platform
 - Windows or Linux-based RabbitMQ cluster
 - Or RabbitMQ as a service via CloudAMQP (really good; see <https://www.cloudamqp.com/>)
- Applicable to a wide variety of enterprise integration scenarios
 - Viable replacement for proprietary message queuing solutions
 - Flexible
 - Straightforward to integrate with legacy application code
 - Can also be used with languages such as Ruby, Python, and Java on OpenVMS

RabbitMQ and OpenVMS

www.vmssoftware.com

Publish a message using Python:

```
import pika

connection = pika.BlockingConnection(pika.ConnectionParameters(host='localhost'))
channel = connection.channel()

channel.queue_declare(queue='hello')

channel.basic_publish(exchange='',
                    routing_key='hello',
                    body='Hello World!')
print(" [x] Sent 'Hello World!'")
connection.close()
```

RabbitMQ and OpenVMS

www.vmssoftware.com

Consume messages using Ruby:

```
require 'bunny'

connection = Bunny.new(automatically_recover: false)
connection.start

channel = connection.create_channel
queue = channel.queue('hello')

begin
  puts ' [*] Waiting for messages. To exit press CTRL+C'
  queue.subscribe(block: true) do |_delivery_info, _properties, body|
    puts " [x] Received #{body}"
  end
rescue Interrupt => _
  connection.close

  exit(0)
end
```

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

MQTT

www.vmssoftware.com

- Open Standard; ISO/IEC 20922:2016
 - Invented by Andy Stanford-Clark of IBM, and Arlen Nipper of Cirrus Link Solutions
 - Donated to Eclipse “Paho” M2M project
 - <http://www.eclipse.org/paho/>
- Lightweight publish/subscribe messaging protocol
 - TCP-based (typically)
 - Asynchronous bi-directional push (no polling)
 - Simple, small number of verbs
 - Protocol compressed into bit-wise headers and variable length fields
 - Smallest packet size 2 bytes
 - Small client footprint
 - Reliability levels, session awareness
 - Data-centric, payload-agnostic
 - Separates data (payload) from metadata (topic)



MQTT

www.vmssoftware.com

- Facilitates the transfer of telemetry-style data to a server or message broker from pervasive devices over high-latency or otherwise constrained networks
 - Sensors and actuators
 - Mobile phones
 - Embedded systems on vehicles
 - Laptops and other computing devices

MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over occasional dial-up connections with healthcare providers, and in a range of home automation and small device scenarios. It is also ideal for mobile applications because of its small size, low power usage, minimized data packets, and efficient distribution of information to one or many receivers...

<http://www.mqtt.org>

MQTT history

www.vmssoftware.com

MQTT's origins go back nearly two decades. Its co-inventor, Andy Stanford-Clark (IBM Distinguished Engineer) has long been a passionate home-automation tinkerer. His laboratory has been his house, a 16th-century cottage on the Isle of Wight, and his electronic gadgets range from temperature and energy monitors to an automated mousetrap. His home automation projects required machine-to-machine data communication, and for this purpose Andy wrote his own code.

At IBM Andy became immersed in the technology for machine-to-machine communication in the late 1990s, when IBM were working with industry partners to mine sensor data from offshore oil rigs for preventive and predictive maintenance. One of those industry partners was Arlen Nipper, an American engineer and expert in embedded systems for oil field equipment. Together they wrote the initial version of MQTT in 1998, based on Andy's original home automation code.



MQTT on OpenVMS

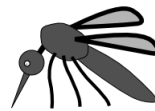
www.vmssoftware.com

Mosquitto MQTT broker:

- <http://mosquitto.org/>
- Open source MQTT broker that implements MQTT protocol versions 3.1 and 3.1.1
- Reference implementation

- OpenVMS port includes all functionality
 - SSL/TLS support
 - Bridging
 - Client authentication
 - Access controls

- SYSUAF authentication plugin



MQTT on OpenVMS

www.vmssoftware.com

Paho MQTT client library:

- <https://www.eclipse.org/paho>
- Fully-featured MQTT client written in ANSI standard C
- Synchronous and asynchronous variants
- OpenVMS port includes all API functionality including SSL/TLS support
- Simple abstraction layer to make it easier to use the API with languages other than C (WIP)



Other:

- Can also use MQTT client libraries for several other languages
 - Java, Lua, PHP, Ruby

MQTT on OpenVMS

www.vmssoftware.com

- Primarily aimed at Internet of Things applications
- SCADA
- Manufacturing
- ...
- Anywhere where lightweight, efficient, and reliable pub/sub is required

- Straightforward to integrate with legacy application code
- Can also be used with languages such as Ruby, PHP, Python, Lua, and Java on OpenVMS

- OpenVMS seems well-suited to the Internet of Things
 - High availability
 - Clustering
 - Security
 - ...
 - MQTT and OpenVMS could be an ideal combination for many Internet of Things-related solutions

Agenda

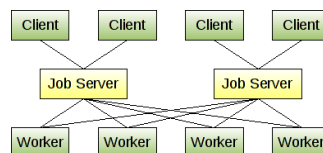
- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

Gearman

www.vmssoftware.com

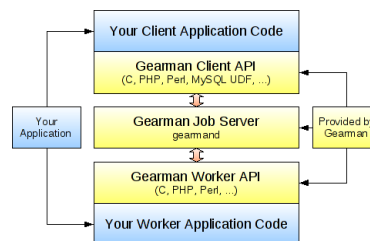
- <http://gearman.org/>
- A generic Open Source application framework to farm out work to other machines or processes that are better suited to perform the work in question
- Several implementations
- Anagram for “manager”
 - Assigns tasks but does none of the real work
- Facilitates doing work in parallel, load balancing of processing, and to calling of functions between languages
- C/C++, Perl, Node.js, Python, PHP, Ruby, Java, .NET, ...
- Applicable to a wide variety of applications
 - High-availability web sites
 - Transport of database replication events
 - ...



Gearman

www.vmssoftware.com

- Highly flexible, scalable, fast, and fault-tolerant
- Persistence (various storage options)
- Natural load distribution
 - Load balance large numbers of calls
 - Easy to scale out



- Clients create jobs to be run and then send them to a job server
- Workers register with a job server and grab (are handed) jobs as they come in
- The job server coordinates the assignment of jobs from clients to workers and handles restarting of jobs if workers go away
- Straightforward to introduce into existing and potentially heterogeneous application environments to facilitate distributed or parallel processing tasks
- Can be used with a wide variety of programming languages

Gearman on OpenVMS

www.vmssoftware.com

- Current release of Gearman for OpenVMS is based on the Gearman 1.1.17 distribution
- Includes all standard functionality
- Additionally includes a language-agnostic API
 - Simplifies writing Gearman applications using 3GL languages such as COBOL, FORTRAN, and Pascal
 - Currently incomplete, although sufficient for most purposes

Gearman on OpenVMS

www.vmssoftware.com

- A simple and convenient means of distributing work
- Asynchronous processing tasks
- Work queues
- Very flexible
- Numerous use-cases

- Straightforward to incorporate into legacy environments
- Language agnostic
- Efficient
- Fits well with the OpenVMS cluster model (although not currently cluster-aware)
- ...

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

Redis

www.vmssoftware.com

- <http://redis.io/>
- Networked, in-memory, persistent, journaled, key-value data store
 - Also supports publish/subscribe
 - Used as a database, cache and message broker
 - Primarily a cache
- Typically holds the entire dataset in memory
- Very fast
- Supports various data structures
 - Strings
 - Hashes
 - Lists and sets
 - Sorted sets with range queries
 - Bitmaps
 - Geospatial indexes with radius queries

Redis

www.vmssoftware.com

- Supports various high-level atomic server-side operations
 - Intersection, union, and difference between sets
 - Sorting of lists, sets, and sorted sets
- Built-in master/slave replication facilities (redundancy)
- Lua scripting
- LRU eviction of cache entries
- Transactions and different levels of on-disk persistence
 - Snapshotting
 - Journaling
- High availability and clustering via extensions
- Bindings for most programming languages
 - C/C++, .NET, Erlang, Java, JavaScript, Lua, Perl, PHP, Python, Ruby, ...

Redis on OpenVMS

www.vmssoftware.com

- OpenVMS port of Redis 4.0.0
 - Alpha and Integrity
- Includes all in-memory/caching functionality, Lua scripting, and publish/subscribe functionality
- Language-agnostic API
 - Straightforward to develop applications using 3GL languages such as COBOL, Pascal, and FORTRAN
- PHP extension
- On-disk persistence not currently supported (working on it)
- Uses 32-bit pointers
 - Limits maximum cache size to around 980MB
 - Future releases will use 64-bit pointers

Redis on OpenVMS

www.vmssoftware.com

- General purpose caching
 - Frequently accessed static application data
 - Improving performance of web applications
 - ...
- In-memory database
- Basic pub/sub functionality
- Straightforward to integrate with legacy application code
- PHP extension
- Bindings for other languages will also work on OpenVMS

Redis on OpenVMS

www.vmssoftware.com

Publish to a topic from FORTRAN:

```
program publish
implicit none
include          'redis$root:[include]redisdef.for'

integer*8       ctxt
integer*8       reply
character*32     msg

call redis$connect(%ref(ctxt), %descr("127.0.0.1"), %val(6379))

write(msg, '(a,a)') 'Hello', char(0)
call redis$command(%val(ctxt), %ref(reply), %descr('PUBLISH test %s'), %ref(msg))

call redis$free_reply_object(%val(reply))
call redis$free(%val(ctxt))

end
```

Redis on OpenVMS

www.vmssoftware.com

Populate and retrieve a list using PHP:

```
<?php
$redis = new Redis();
$redis->connect('127.0.0.1', 6379);
echo "Connection to server sucessful";

// Store list data
$redis->lpush("language-list", "COBOL");
$redis->lpush("language-list", "FORTRAN");
$redis->lpush("language-list", "Pascal");

// Get the stored data and print it
$arList = $redis->lrange("language-list", 0 ,5);
print_r($arList);
$redis->delete("language-list");
?>
```

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

WSIT

www.vmssoftware.com

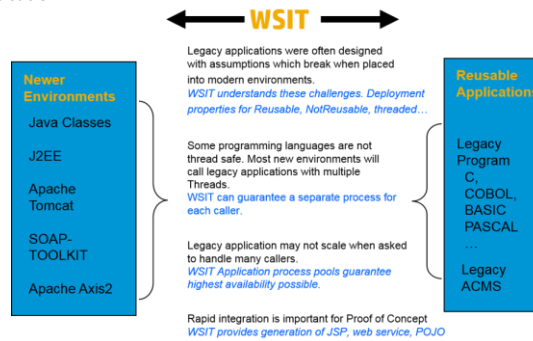
- Web Services Integration Toolkit (WSIT)
- Has nothing to do with web services
 - Web services are supported by Apache AXIS2 and Tomcat
- Not really open source, but it's free ☐

- Extensible API-level integration technology for OpenVMS
- Facilitates the integration of new or existing code written in 3GL languages with Java
- Generates Java beans and C language wrapper code that can be used to construct interfaces to 3GL code and to ACMS
 - Powerful code generation facility
 - Uses Apache Velocity (see <https://velocity.apache.org/>)
 - Velocity templates for a variety of scenarios
 - Can define your own templates
- Generated Java code can be incorporated into any type of Java application

WSIT

www.vmssoftware.com

- Tools understand OpenVMS data types and argument passing mechanisms
- Runtime understands concepts such as processes, threading, and inter-process communication on OpenVMS
- ICC-based IPC or direct JNI calls
- Process/server pooling
- SYSUAF-based and ACMS authentication
- ...



Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

SOAP-based web services

www.vmssoftware.com

- Lightweight (!?) protocol based on XML as the marshalling format for data in request and response messages
- Encoding rules for data type instances
- Vendor and platform-neutral
- Programming language neutral
- Transport neutral
- Designed for loosely-coupled distributed computing



RESTful web services

www.vmssoftware.com

- *Representational state transfer* (REST)
 - A fancy (and somewhat miss-used) term for a fairly simple concept
 - Introduced and formalized by Roy Fielding (2000) in his doctoral dissertation "*Architectural Styles and the Design of Network-based Software Architectures*"
 - <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- An architectural style that is most often applied to the development of web services
 - The World Wide Web effectively conforms to the REST architectural style

The concept:

Representational State Transfer is intended to evoke an image of how a well-designed Web application behaves: presented with a network of Web pages (a virtual state-machine), the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use.

REST was initially described in the context of HTTP, but it is not limited to that protocol. RESTful architectures may be based on other Application Layer protocols if they already provide a rich and uniform vocabulary for applications based on the transfer of meaningful representational state. RESTful applications maximize the use of the existing, well-defined interface and other built-in capabilities provided by the chosen network protocol, and minimize the addition of new application-specific features on top of it.

http://en.wikipedia.org/wiki/Representational_state_transfer

RESTful web services

www.vmssoftware.com

- API's adhere to the following characteristics...
 - A base URI/URL
 - Use an internet media type (typically JSON or XML) for request and response data
 - Use of standard HTTP verbs/methods for certain types of operation (see table below)
 - Use of hypertext links to reference state
 - Use of hypertext links to reference related resources

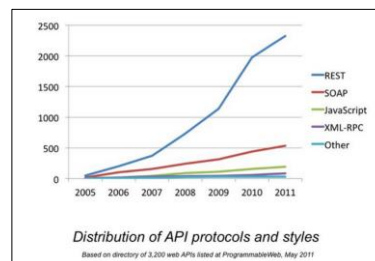
Resource	GET	PUT	POST	DELETE
Collection URI, such as http://example.com/resources	List the URIs and perhaps other details of the collection's members.	Replace the entire collection with another collection.	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	Delete the entire collection.
Element URI, such as http://example.com/resources/item17	Retrieve a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	Replace the addressed member of the collection, or if it doesn't exist, create it.	Not generally used. Treat the addressed member as a collection in its own right and create a new entry in it.	Delete the addressed member of the collection.

http://en.wikipedia.org/wiki/representational_state_transfer

RESTful web services

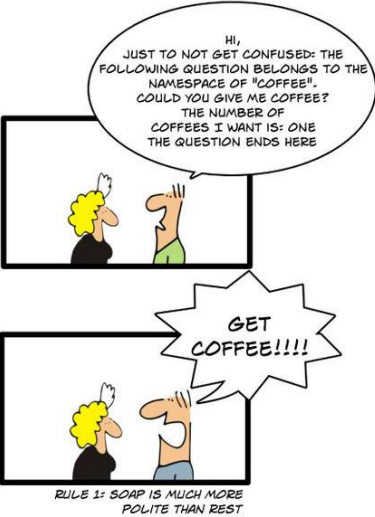
www.vmssoftware.com

- No official standard for RESTful services
 - REST is an architectural style
 - SOAP is a defined standard (large set of standards)
- Wide-spread adoption
 - Far more common than SOAP-based services
 - Most cloud-based services provide some form of RESTful interface
 - Many software products provide RESTful interfaces for various purposes



Comparing SOAP-based and RESTful services

www.vmssoftware.com



gSOAP

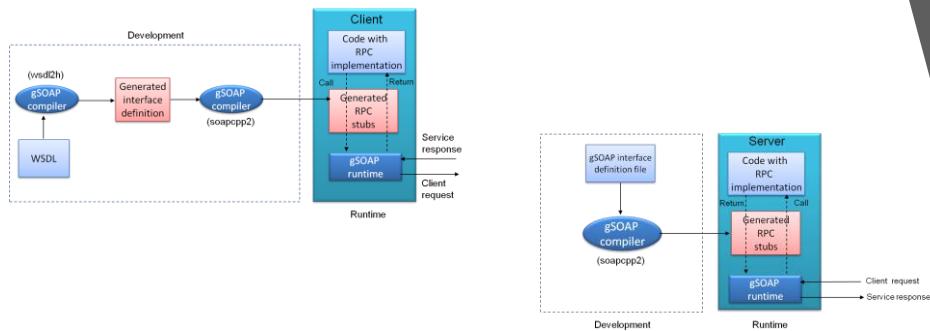
www.vmssoftware.com

- Full-featured Open Source SOAP technology/toolkit
- <http://www.cs.fsu.edu/~engelen/soap.html>
- Thousands of registered users
- Uses a source-to-source stub and skeleton compiler to automate the integration of SOAP RPC in applications
 - Client and server development
 - Automates the deployment of (legacy) 3GL applications as Web services
 - Primarily intended for use with C/C++, but can be used (with a little effort) with any 3GL
- Very fast by web service technology standards
 - Generates fast (de)marshalling routines and streaming XML parsers
 - Efficient run-time
- Application-centric
 - Minimize legacy application code adaptation
 - Support (de)marshalling of application's native data structures in SOAP/XML
 - Preserve the logical structure of data

gSOAP

www.vmssoftware.com

- Major components available for OpenVMS (Alpha and IA64)
 - Stub compiler, WSDL parser, runtime, ...
 - Various extensions to simplify use from languages other than C/C++
 - ACMS support (threaded agent gateway, tools to assist with interface creation)
 - Integration with Apache via `MOD_GSOAP`
 - Numerous OpenVMS users (25+)



gSOAP – what's it being used for?

www.vmssoftware.com

- In excess of two dozen organizations using gSOAP on OpenVMS (Alpha and Integrity) in production environments
- Multiple sectors
 - Finance
 - Healthcare
 - Telecommunications
 - Tourism
 - Manufacturing
 - Entertainment
 - ...
- Fulfilling business-critical functions
- Integration with OpenVMS applications written in various languages and technologies
 - C, C++, COBOL, FORTRAN, BASIC, Pascal
 - ACMS, DEC MessageQ, ...

gSOAP – what's it being used for?

www.vmssoftware.com

- Calling remote/external web services from OpenVMS
 - Maybe a business partner, third-party service, or internal BU has a service you need to access
- Providing web services on OpenVMS
 - Provide a standards-based client-server style interface to new or existing applications
- Ease of integration with existing 3GL code (and ACMS)
- Good performance and stability
- Lightweight
- Flexible
- Excellent interoperability with other web services technologies
- No separate runtime to install or manage

gSOAP – what's it being used for?

www.vmssoftware.com

Some examples:

- A large medical insurance company uses gSOAP with Apache HTTPD to provide a web services interface to their legacy ACMS environment
- A large healthcare provider uses gSOAP on OpenVMS to integrate a .NET application used by doctors across the US with their back-end systems
- A steel manufacturer uses gSOAP to communicate between their OpenVMS based systems and factory cranes fitted with on-board computers running a .NET application
- A UK-based ticketing company uses gSOAP to provide a web services interface to their ACMS application and to call web services from COBOL code within their applications
- And many more

RESTful interfaces on OpenVMS

www.vmssoftware.com

- Numerous frameworks in various languages
- Jetty
 - Java
 - <http://www.eclipse.org/jetty/>
 - Can be used on OpenVMS with Java 8
- Ruby Sinatra
 - <http://www.sinatrarb.com/>
 - Works well on OpenVMS
- Eve
 - Python
 - <http://python-eve.org/>
 - Probably works on OpenVMS (have not tried it)
- Many more

- But these frameworks cannot necessarily easily be hooked into existing 3GL code
 - May be possible to do something with Swig
 - Or WSIT (Java only)

```
require 'sinatra'
get '/' do
  "Just Do It"
end
```

RESTful interfaces on OpenVMS

www.vmssoftware.com

- An embedded web server like Mongoose works well on OpenVMS for the development of RESTful web services
 - <https://cesanta.com/>
- gSOAP includes a comprehensive and efficient HTTP library
 - Fast and simple to use
 - Currently limited server-side support for RESTful services via a plugin
- libcurl provides comprehensive HTTP(s) client support
 - Arguably the best library to use on OpenVMS for implementing interfaces to RESTful
 - Can be integrated into existing 3GL code
 - I've used this to develop interfaces to various services
- Most RESTful web services send/return either JSON or XML requests/responses
 - Not strictly a requirement
 - Sometimes both formats are supported and you can specify which you want
 - Need either an XML or a JSON parser/API (no problem)

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

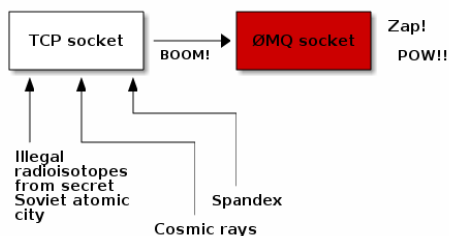
Other useful stuff

www.vmssoftware.com

ZeroMQ (efficient and coordinated IPC):

- A high-performance, low-latency messaging platform
- <http://www.zeromq.org>
- Easy-to-use API
- Great documentation
- Supports multiple protocols
- TCP, PGM (multicast), SCTP, in-process, and TIPC
- Fully distributed
- Innumerable messaging/IPC patterns/topologies
- Runs on numerous platforms
- Language bindings available for just about any language you can think of
- Build available for OpenVMS

A ØMQ socket is what you get when you take a normal TCP socket, inject it with a mix of radioactive isotopes stolen from a secret Soviet atomic research project, bombard it with 1950-era cosmic rays, and put it into the hands of a drug-addled comic book author with a badly-disguised fetish for bulging muscles clad in spandex. (Pieter Hintjens, 2011)



Other useful stuff

www.vmssoftware.com

HAProxy (community edition):



- <http://www.haproxy.org/>
- Offers high availability, load balancing, and proxying for TCP and HTTP-based applications
- Particularly suited for high traffic web sites
- Easy to use and to integrate into existing environments
- Not exactly an integration technology, but part of the picture

Apache ActiveMQ:



- <http://activemq.apache.org/>
- Implementation of the Java Message Service specification
- Provides support for JMS and various other messaging protocols, including AMQP 1.0, MQTT, and STOMP, as well as its native OpenWire protocol
- Integrates well with Java EE containers, ESBs, and other JMS providers

Other useful stuff

www.vmssoftware.com



librdkafka:

- <https://github.com/edenhill/librdkafka>
- A C API for interacting with Apache Kafka
 - Apache Kafka is an open source message broker written in Java and Scala
 - <https://kafka.apache.org/>
 - Aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds
- Producer and consumer support
- Provides a portable C/C++ API that can be used by client applications to interact with the message broker

- OpenVMS port includes all functionality
- Also provides a wrapper API that makes it easier to use the API with other OpenVMS 3GL's such as COBOL, FORTRAN, Pascal, and BASIC

Agenda

- ▶ Introduction
- ▶ RabbitMQ (AMQP, MQTT, STOMP, ...)
- ▶ MQTT
- ▶ Gearman
- ▶ Redis
- ▶ WSIT
- ▶ Web services (SOAP and REST)
- ▶ Other stuff
- ▶ Summary

www.vmssoftware.com

Summary

www.vmssoftware.com

- Open source integration solutions abound for OpenVMS
- And we'll probably add a few more
- From an integration perspective you can do on OpenVMS pretty much anything you can do on other platforms
 - And it can be done cost-effectively
- We considered a few options in this talk
 - Somewhat different characteristics
 - The notion of a “one-size-fits-all” integration is flawed
 - Need to carefully consider your requirements

Summary

www.vmssoftware.com

- Use of open standards is important
 - Flexibility around product choices
 - Reduces lock-in
 - Interoperability
 - Availability of skills
 - Reduces costs
 - ...
- Integration is fun ☺
 - But all integration projects are different
 - If you need help, contact us



www.vmssoftware.com

Questions?

