RestMS

A new protocol for
web messaging

by Pieter Hintjens, iMatix Corporation

February, 2009

iMatix Corporation

Why do we need Messaging?

- To connect the pieces together
  - It's cheap to make applications
  - But how do they talk to each other?
- To create information feeds
  - “I'm renting my apartment”
- To mine information feeds
  - “I'm searching for an apartment”
  - Think: Yahoo! Pipes

iMatix Corporation

What Messaging should offer

- Easy to connect to from any application
- Generic tools for creating & mining feeds
- Early filtering vs. late filtering
  - Don't send unwanted data over the network
- Scalability
  - 1, 10, 100, 1000, 10K, 100K messages/sec
- Loose coupling
  - Hiding applications from each other

iMatix Corporation

Use cases

- “Server-side search of current events”
- Watch IRC channels for interesting stuff
- Monitor thousands of news channels
- Track thousands of postings and uploads
- Follow thousands of stock values
- etc.

iMatix Corporation

The State of the Art

- Web messaging
  - Simple direct messaging
  - RSS, AtomPub, XMPP, SOAP
- Business messaging
  - Java repackaging of legacy tools
  - Queues, pub-sub
- Enterprise messaging
  - Mostly expensive closed products
  - Queues, pub-sub

iMatix Corporation

Challenges

- Routing model
  - Hard-wired or dynamic?
  - Fixed algorithms or extensible?
- Protocol vs. product
  - Real competition in the market?
  - Free and open digital standard?
- Complexity
  - Simple to understand and use
  - Or, “enterprise technology”?

iMatix Corporation

Why is messaging unpopular?

- Too complex
  - “Results 1 - 10 of about 264,000 for 'soap too complex'.”
  - Complexity means extra cost
- Not interoperable
  - Products are not standards
  - Languages are not universal
- Still too few FOSS solutions
  - Mainly due to lack of open standards

iMatix Corporation

In an Ideal World...

- Based on free and open standards
- Generic & extensible routing
- Easy to use in any language
- Runs nicely on any operating system
- Fast enough for real work
  - Low latency (10usec to 1sec)
  - High volume (1/sec to 1M /sec)

iMatix Corporation

AMQP

- ✔ Free and open standard
- ✔ Generic & extensible routing
- ✗ *Not easy to use in any language*
  - ✗ *It's a complex binary protocol that needs an API stack for each language*
- ✔ Runs nicely on any operating system
- ✔ Fast enough for real work

- AMQP is great except it's inaccessible

iMatix Corporation

AtomPub

- ✔ Free and open standard
- ✗ *A single limited routing model*
  - ✗ *Publish to feed, read from feed: no filtering or routing*
- ✔ Easy to use in any language
- ✔ Runs nicely on any operating system
- ✔ Fast enough for real work

- AtomPub is great except it's limited

iMatix Corporation

So why RestMS?

- AMQP showed us something new
  - Generic routing (“exchange-binding-queue”)
  - Portability (Linux, AIX, Solaris, win32)
  - Interoperability (C, C++, Java, .NET, Python)
  - Performance (~500k msg/sec in OpenAMQ)
- But it remains inaccessible
  - How do I use AMQP from the web?
  - How do I use AMQP in language X?

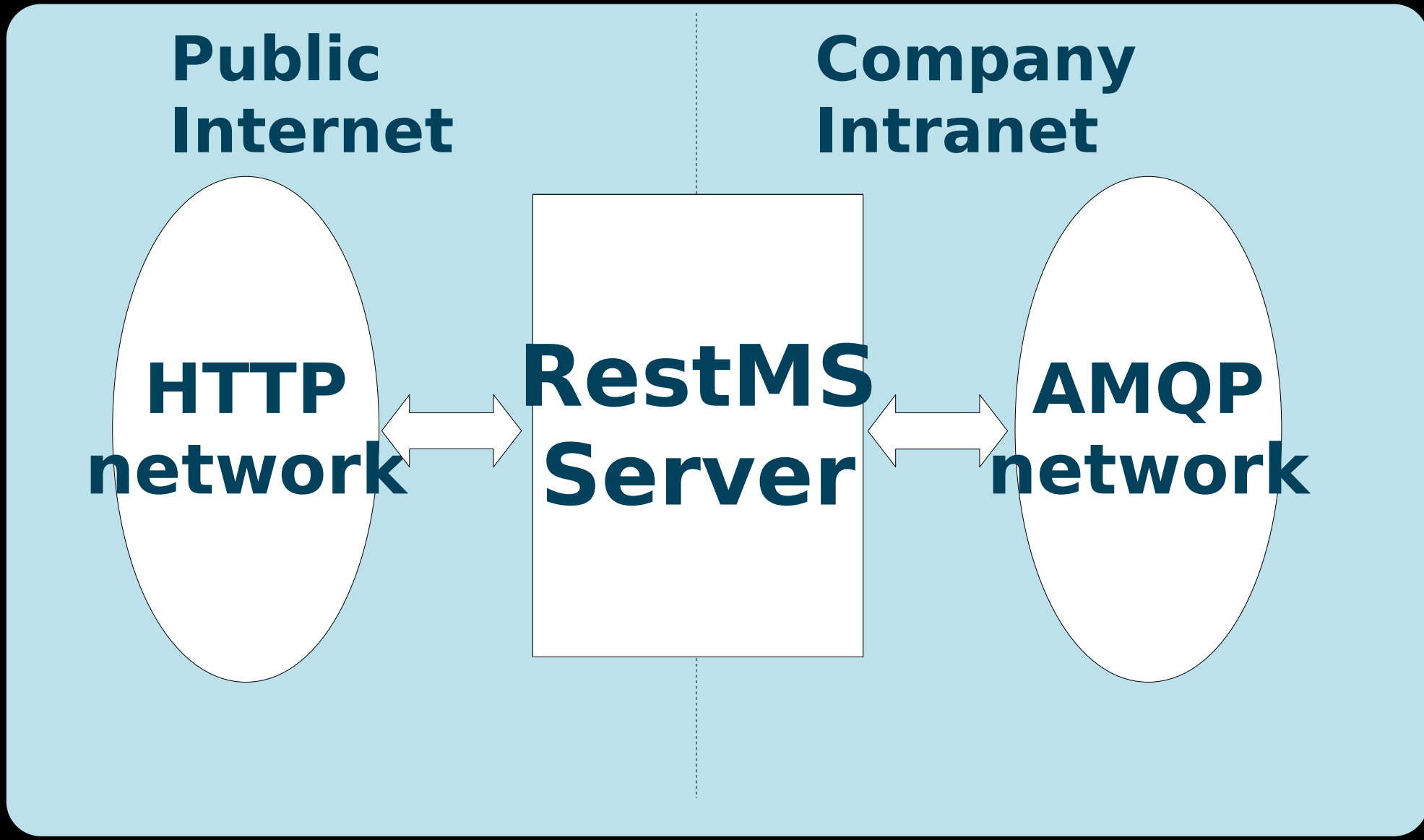
iMatix Corporation

Magic potions

- Take the best of AMQP and AtomPub
- Shake briskly, allow to cook:
- AMQP's routing model (improved)
- RESTful access over ordinary HTTP(S)
- Portable & interoperable
- Documented as free and open standard
- Interoperate with AMQP networks

iMatix Corporation

Architecture



iMatix Corporation

How RestMS works

- Publishers send messages to *feeds*
- Subscribers create *pipes*
  - And *join* their pipes to feeds
- Subscribers then read from their pipes
  - Get message, process message, loop
- Joins specify the routing arguments
  - By key, by expression, etc.

iMatix Corporation

RESTful access

- If you know HTTP, you know REST
- Create/Read/Update/Delete resource
  - POST creates a new resource
  - GET reads a resource
  - PUT updates a resource
  - DELETE spawns... just kidding, it deletes the resource
- Resources are mostly shown in XML
  - But RestMS also supports JSON

iMatix Corporation

Why REST / HTTP is cool

- It is simple
- It uses existing transports
- It uses existing security
- It uses existing libraries
- It uses existing metaphors (CRUD)
- It scales on a web infrastructure
  - Proper attention to caching

iMatix Corporation

REST / HTTP gotchas

- Rather chatty
  - Can be solved by batching writes and reads
- Polled vs. event driven
  - Can be solved by using “long polls”
- Rather verbose (XML envelopes)
  - Ignore the problem, it will go away
- Not an “enterprise technology”
  - Wait, it will be

iMatix Corporation

What's available now?

- **restms.org**
  - Draft RestMS specifications
  - Example client code (Perl, for now)
- **openamq.org**
  - Source code for OpenAMQ & Zyre
- **zyre.com**
  - All about Zyre
- **live.zyre.com**
  - Live RestMS server, running Zyre

iMatix Corporation

Conclusions

- Messaging is a great tool
- But the good stuff is too complex
- And the simple stuff is too limited
- Time for a simple, capable option
- Based on a free and open standard
- With a solid implementation

iMatix Corporation

Thank you

- For more information please contact the author at ph@imatix.com
- Comment & discuss at zyre.com
- Email list, see www.openamq.org

iMatix Corporation