Prosody as a Jabber Server and for Generic XML Routing

Matthew Wild
matthew@prosody.im

Prosody

- Project started Summer 2008
- Written in Lua, C
- < 200KB in size, Lua itself around the same
- Small runtime footprint too
- Scales
  - From routers...
  - ...to large-scale services

XMPP

- Began life in 2000 as Jabber

- IETF standard (RFCs 3920, 3921)

- XMPP Standards Foundation

- Extensions (lots of them!)

- Core: messaging ("Hi!"), presence ("I'm online")

XMPP: IM

- Typical use... instant messaging
- Federated – architecture similar to email
    - Open federation – run your own server
    - Maintains "strong identity"
- Prosody popular on personal servers
- Though also caters for more advanced usage

"Enterprise" IM

- Shared groups

  - Central management of contact lists, who sees who

- Message logging

- Policy enforcement

  - Encryption

- Chat rooms

- Website integration

- Cyrus SASL and other authentication backends

Beyond IM

- XML - extensible!

```
<message to="alice@wonderland.lit" from="bob@realworld.irl">
   <body>Hello!</body>
</message>
```

```
<message to="alice@wonderland.lit" from="bob@realworld.irl">
   <body>Hello!</body>
   <greet xmlns="http://example.com/greetprotocol">
      <type>wave</type>
   </greet>
</message>
```

XMPP on the web

- Bidirectional-streams Over Synchronous HTTP
  - Similar to AJAX "long-polling", there is always a request open for the server to send data on
  - Client can make new request at any time to send data (server replies to the original request)
  - Javascript libraries: Strophe.js, JSJaC, ...
  - Native BOSH support in many XMPP servers
- Flash, websockets...
- A world of possibilities...

Example use: Monitoring

- Personal eventing protocol ("PEP")
    - Publish data to your account (just like presence)
    - Except recipents don't receive it unless they ask
- Multiple publishers (the monitors)
    - Daemon publishes values when they change
- Multiple subscribers (the viewers)
    - Desktop app, web app, mobile app...
    - Automatically receive updates they ask for

XMPP Monitoring Demo

- Proof-of-concept: http://monitor.prosody.im/

- Javascript client (Strophe.js XMPP library)

- Lua script monitors system values via /proc and e.g. Munin plugins, publishing changes

- Prosody acts as a smart router for all this data

  - Stores last values for sending to new viewers

  - Filter broadcasts to viewers by request (mobile client would receive critical alerts only, etc.)

Thank you!

- Questions?

- Links

  - Website: http://prosody.im/

  - Blog: http://blog.prosody.im/

  - Email: matthew@prosody.im

  - Twitter: http://twitter.com/prosodyim