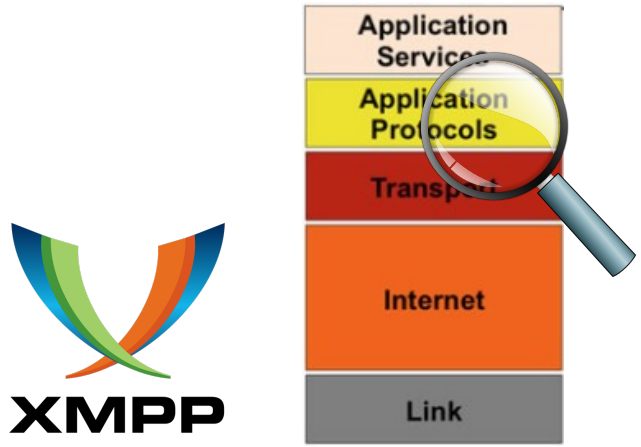
Ch. 13 - IoT Application Protocol Layer

Sec 3 – XMPP Protocol

COMPSCI 147

Internet-of-Things; Software and Systems



APP PROTOCOLS FOR IOT - STANDARDIZATION

- **HTTP**
  - **IETF standard (RFC 2616 is HTTP/1.1)**
- **CoAP**
  - **IETF standard (RFC 7252)**
- **XMPP**
  - **IETF standard (RFC 6272)**
- **MQTT**
  - **OASIS standard**
- AMQP
  - OASIS and ISO 19464 standard (1.0)
- SIP
  - IETF Standard (RFC 3261)
- IEEE 1888
  - IEEE Standard
- DDS (RTPS)
  - Object Management Group (OMG) Standard

WHAT IS XMPP?

- XMPP stands for eXtensible Messaging and Presence Protocol

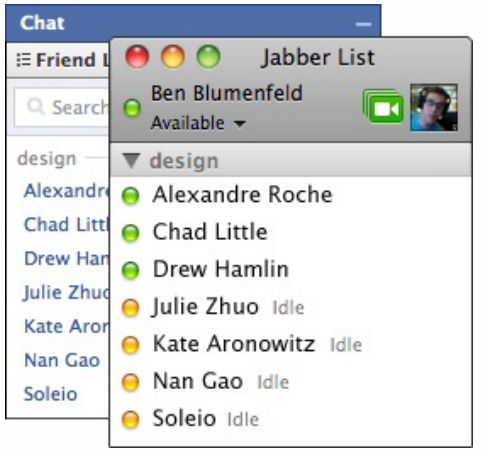
  Developed as instant **messaging** (IM) open-source used by ICQ, AIM, and MSN in 1999 (short messaging client-client / client-server)

  – **eXtensible**: It can be customized to several needs, including M2M communication.

  – **Messaging**: Primary method of communication of short messages.

  – **Presence** Logic: Presence status con trigger customizable events.

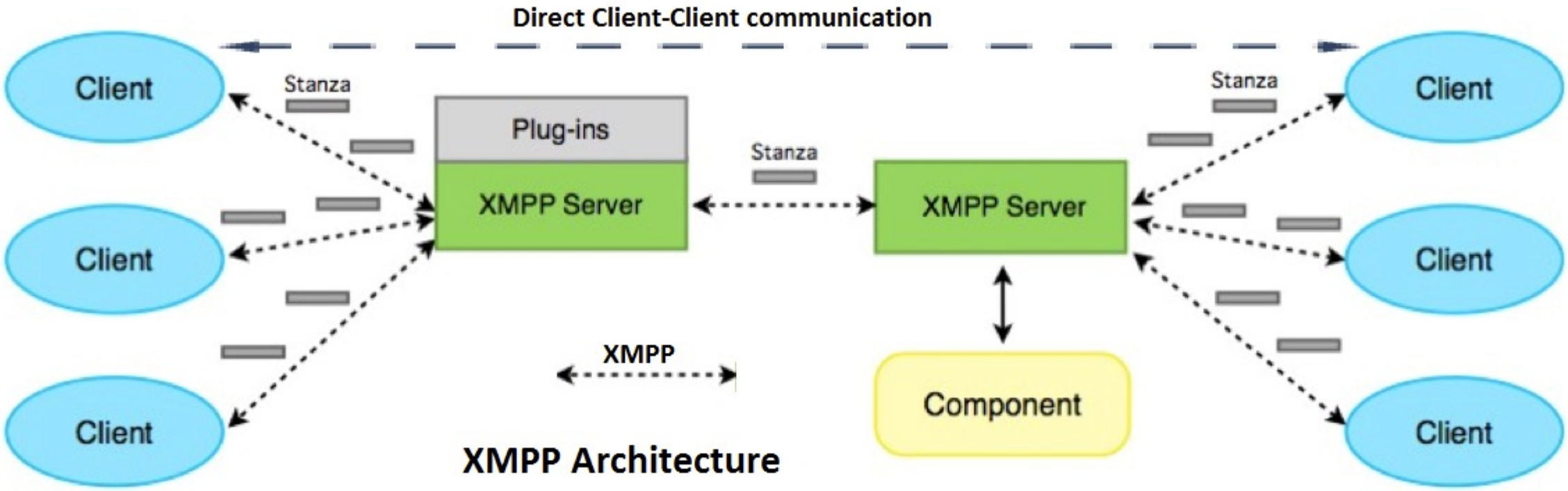
  – Open **Protocol** for Bi-directional streaming XML based.



- Originally referred as Jabber

WHO USES/USED XMPP?

XMPP ARCHITECTURE



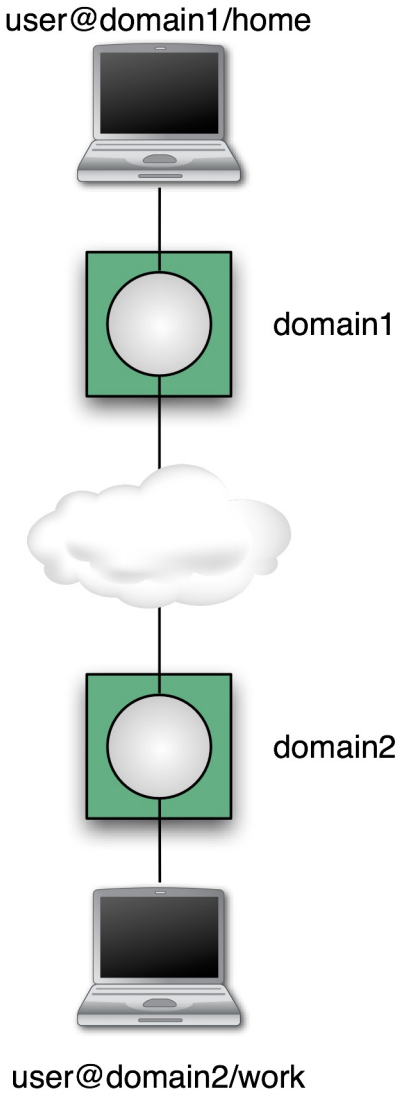
XMPP Architecture

XMPP ARCHITECTURE

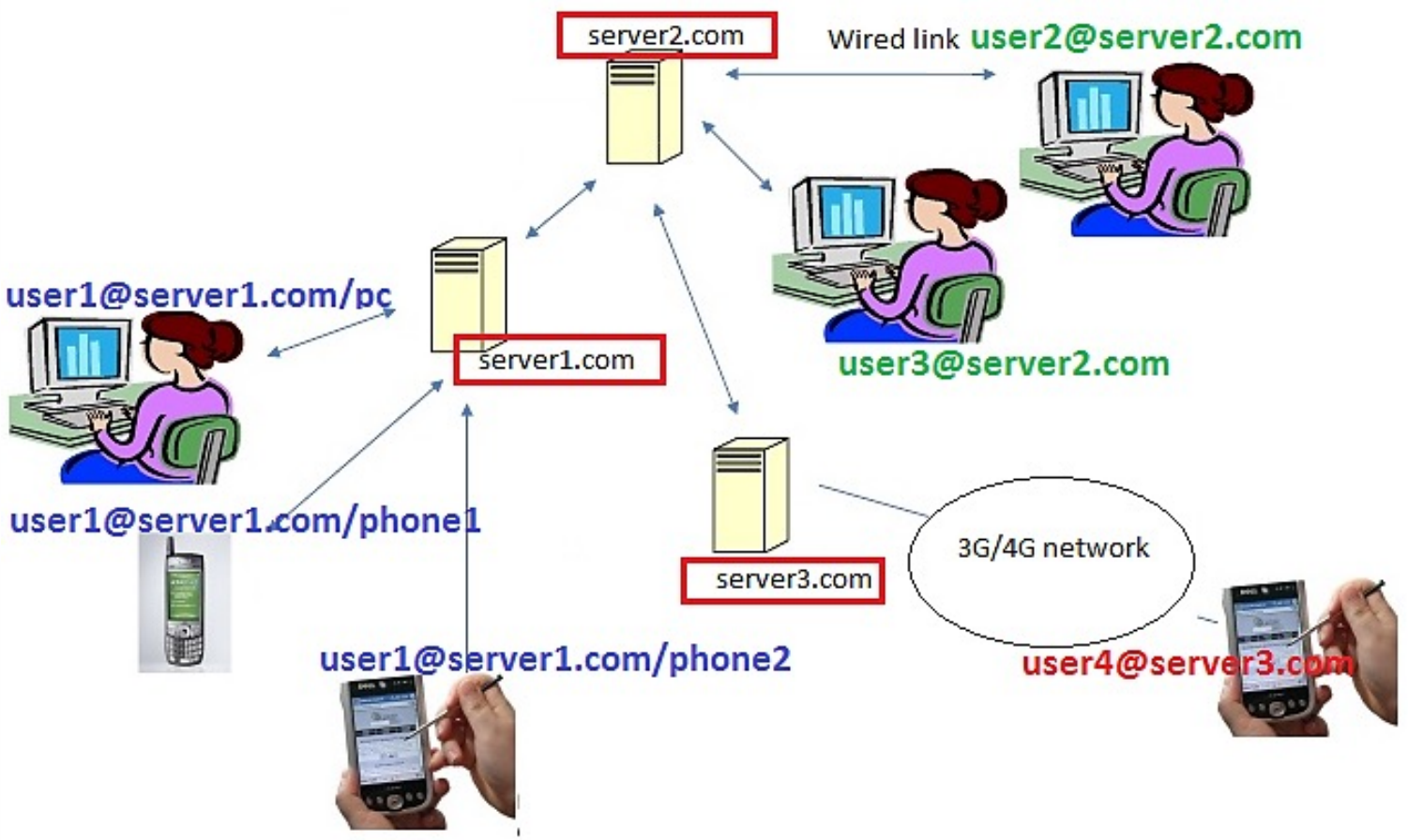
- Addressing Scheme:
  JID = Jabber ID *(Jabber is the original name of XMPP)*= node@domain/resource
  - Node: identity, e.g. username
  - Domain: DNS domain name (IP)
  - Resource: device identifier

Identifies a person

- Client talks to "local" server
  - Wherever the user account is hosted
  - Tied to directory if desired
  - Organizational policy enforced
- Servers talk to other servers
  - DNS lookup on domain portion of address



XMPP ADDRESSING



XMPP STREAMS

- Client connects TCP socket to server

- Client sends stream start tag:

```
<stream:stream xmlns='jabber:client'
        xmlns:stream='http://etherx.jabber.org/streams'
        to='example.com'
        version='1.0'>
```

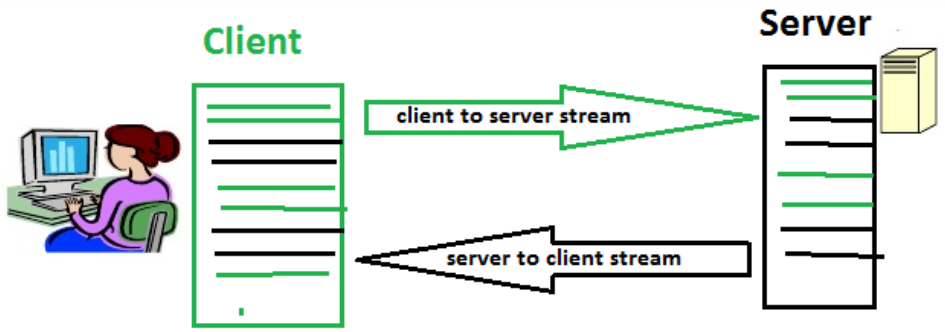
NOT an element

- Server sends stream start tag back:

```
<stream:stream xmlns='jabber:client'
        xmlns:stream='http://etherx.jabber.org/streams'
        from='example.com'
        id='someid'
        version='1.0'>
```

NOT an element

- Each child element of stream is a "*stanza*"



Communication is through stanzas

The unit of communication in XMPP is called a stanza

Short Extensible Markup Language (XML) messages

```
<Presence
from="user1@mydomain.com"
id="1232312312312"
to="user2@mydomain.com"
>
  <show>online</show>
</Presence>
```

```
<iq id="9b85:sendIQ" to="user2@mydomain.com" type="get" >
<vCard xmlns="vcard-temp"/>
</iq>
```

```
<message
from="user1@mydomain.com"
id="1232312312312"
to="user2@mydomain.com"
type="chat"
>
<body>Hi friends</body>
</message>
```

XMPP - STANZAS

- XMPP communication happens through stanzas, the basic comm unit.
- Stanzas are short XML messages with to='JID' and from='JID' addresses
  - "To" gives destination
  - "From" is added by server
- Different types for delivery semantics

  <message/>: one direction, one recipient

  <presence/>: one direction, publish to many

  <iq/>: "info-query", request/response

XMPP - STANZAS - MESSAGE

- Example:

```
<message xml:lang='en'
        to='romeo@example.net'
        from='juliet@example.com/balcony'
        type='chat'>
  <body>Wherefore art thou, Romeo?</body>
</message>
```

- Types: chat, group chat, error, ...

- Body: plain text / HTML

XMPP - STANZAS - PRESENCE

- Example:

```
<presence>
  <show>dnd</show>
  <status>Meeting</status>
  <priority>1</priority>
</presence>
```

- Show: How the user status should be presented
  - chat, available, away, xa(away for an extended period), dnd (do not disturb)
- Status: Human-readable text, customizable
- Priority: Defines the availability of the user/device.

XMPP - STANZAS - SUBSCRIBING TO PRESENCE

- Send a subscription request:

```
<presence to='juliet@example.com'
      type='subscribe'/>
```

- Approving a request:

```
<presence to='romeo@example.net'
      type='subscribed'/>
```

- Every time you change a subscription, you get a "roster push":

```
<iq type='set'>
 <query xmlns='jabber:iq:roster'>
  <item jid='romeo@example.net'
      subscription='from'/>
 </query>
</iq>
```

XMPP - STANZAS – IQ REQUEST

- Can request information from the server (e.g. roster)
- Can apply settings to the server (e.g. remove subscriptions)
- Options
  - Type: get, set, result, error
  - ID: track the corresponding response
  - Query/Namespace: what type of request?
- Example: request the roster

```
<iq type='get'
    id='roster_1'>
  <query xmlns='jabber:iq:roster'/>
</iq>
```

XMPP - STANZAS – IQ RESPONSE

- Example: return roster

```
<iq type='result'
    id='roster_1'>
  <query xmlns='jabber:iq:roster'>
    <item jid='romeo@example.net'
          name='Romeo'
          subscription='both'>
      <group>Friends</group>
    </item>
  </query>
</iq>
```

- Type: response
- ID matches request
- Subscription state: none, to, from, both

AN ONLINE SHORT TUTORIAL TO XMPP

- Introduction to XMPP
  - https://www.youtube.com/watch?v=68G4js91xrQ&t=132s

- Popular XMPP servers:
  - ejabberd: https://github.com/processone/ejabberd
  - Openfire: https://github.com/igniterealtime/Openfire
  - Prosody: https://github.com/bjc/prosody

So what happened to XMPP

ejabberd.jabber.narkive.com/Jc6l0B1n/leaking-and-crashing

NARKIVE MAILINGLIST ARCHIVE ejabberd@jabber.ru

Discussion:

[ejabberd] ejabberd leaking and crashing

Jan Koum 12 years ago

hey guys,

i am not sure if we are reaching the limits of ejabberd can do, but hopefully not..

we have about 5,000 connected uses at a time and about 500,000 total registered uses.

ejabberd has slowly been growing its memory usage until twice in the past 12 hours it crashed with:

swap_pager_getswapspace(2): failed
swap_pager_getswapspace(4): failed
Dec 10 08:13:14 im101 last message repeated 37 times
pid 96140 (beam), uid 1000, was killed: out of swap space
Dec 10 08:13:15 im101 kernel: pid 96140 (beam), uid 1000, was killed: out of swap space

machine is FreeBSD 7.2 with 8GB of RAM, ejabberd is 2.1.0-RC2

3 Replies
29 Views
Permalink to this page
Disable enhanced parsing

Thread Navigation

Jan Koum	12 years ago
Jan Koum	12 years ago
Peter Viskup	12 years ago
Peter Viskup	12 years ago

https://ejabberd.jabber.narkive.com/Jc6l0B1n/leaking-and-crashing

So what happened to XMPP

ejabberd.jabber.narkive.com/Jc6l0B1n/leaking-and-crashing

NARKIVE MAILINGLIST ARCHIVE ejabberd@jabber.ru

Discussion:
[ejabberd] ejabberd leaking and crashing

Jan Koum 12 years ago

hey guys,

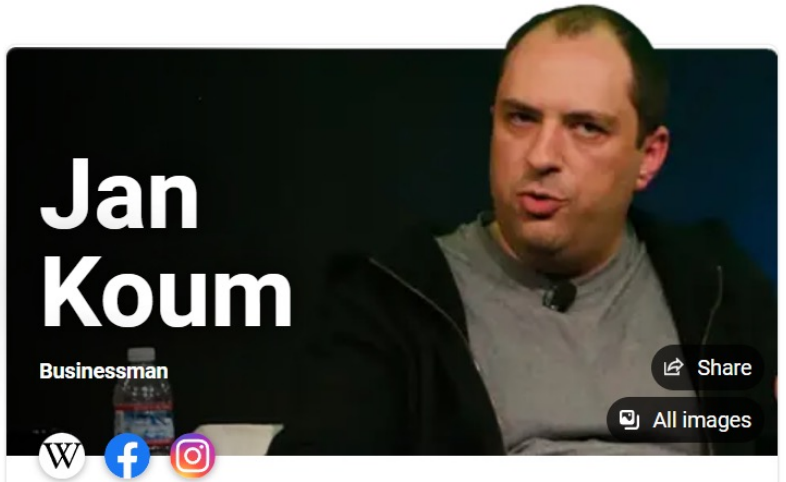
i am not sure if we are reaching the limits of ejabberd can do, but hopefully not..

we have about 5,000 connected uses at a time and about 500,000 total registered uses.

ejabberd has slowly been growing its memory usage until twice in the past 12 hours it crashed with:

swap_pager_getswapspace(2): failed
swap_pager_getswapspace(4): failed
Dec 10 08:13:14 im101 last message repeated 37 times
pid 96140 (beam), uid 1000, was killed: out of swap space
Dec 10 08:13:15 im101 kernel: pid 96140 (beam), uid 1000, was killed: out of swap space

machine is FreeBSD 7.2 with 8GB of RAM, ejabberd is 2.1.0-RC2



Jan Koum is a Ukrainian-American billionaire businessman and computer engineer. He is the co-founder and former CEO of WhatsApp, a mobile messaging app which was acquired by Facebook in 2014 for US$19.3 billion. According to Forbes, he has an estimated net worth of US$10.7 billion as of January 2022, making him one of the richest people in the world. He entered the Forbes list of the 400 richest Americans in 2014 at No. 62, with an estimated net worth of $7.5 billion, the highest-ranked newcomer to the list that year.
Wikipedia

Born Feb 24, 1976 (age 46) · Kiev, Ukraine
Net worth $9.88 billion (2022)
Founded WhatsApp Inc.

https://ejabberd.jabber.narkive.com/Jc6l0B1n/leaking-and-crashing

Cell phones happened !

- Web-scalability: Extremely heavyweight
  16 byte per ASCII character and that in XML!

- Not mobile-phone friendly (drains a lot of battery)
  - CPU usage
  - handling mobile connections flakiness
  - high bandwidth usage of XML stanzas
  - push notifications support

- GPL License (need to open-source every custom module/ improvement)

https://www.youtube.com/watch?v=s1Rd4UShDxQ