

# XMPP FOR DEVELOPERS

Daniel Gultsch

September 17th, 2019 @ XMPP Meetup Dresden

# SLIDES

[github.com/talks/inputmice](https://github.com/talks/inputmice)

# STREAMS

## OUTPUT STREAM (CLIENT TO SERVER)

```
<?xml version="1.0"?>
<stream>

<message to="alice">
  <body>Hi</body>
</message>
<iq to="eve" type="set" id="x">
  <ping xmlns="urn:xmpp:ping"/>
</iq>

</stream>
```

## INPUT STREAM (SERVER TO CLIENT)

```
<?xml version="1.0"?>
<stream>

<iq from="eve" type="result"
  id="x"/>

</stream>
```

# 1. CLIENT TO SERVER

```
<?xml version='1.0'?>
<stream:stream from='d@conversations.im' to='conversations.im'
  version='1.0' xml:lang='en' xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'>
```

# 2. SERVER TO CLIENT

```
<?xml version='1.0'?>
<stream:stream from='conversations.im' to='d@conversations.im'
  version='1.0' xml:lang='en' xmlns='jabber:client' id="ra
  xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls' />
    <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
      <mechanism>PLAIN</mechanism>
    </mechanisms>
  </stream:features>
```

### 3. CLIENT TO SERVER

```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls' />
```

### 4. SERVER TO CLIENT

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls' />
```

### 5. CLIENT ↔ SERVER

TLS Handshake

### 6. CLIENT TO SERVER (RESTART)

```
<?xml version='1.0'?>  
<stream:stream from='d@conversations.im' to='conversations.im'  
  ...  
  xmlns:stream='http://etherx.jabber.org/streams'>
```

## 7. SERVER TO CLIENT (RESTART)

```
<?xml version='1.0'?>
<stream:stream from='conversations.im' to='d@conversations.im'
  version='1.0' xml:lang='en' xmlns='jabber:client' id="ra
  xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
      <mechanism>PLAIN</mechanism>
    </mechanisms>
  </stream:features>
```

## 8. CLIENT TO SERVER

```
<auth xmlns="urn:ietf:params:xml:ns:xmpp-sasl" mechanism="PLAIN"
  ZABzZWNyZXRwYXNzd29yZAAK
</auth>
```

## 9. SERVER TO CLIENT

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl' />
```

## 10. CLIENT TO SERVER (2. RESTART)

```
<stream:stream from='d@conversations.im' to='conversations.im'  
...  
xmlns:stream='http://etherx.jabber.org/streams'>
```

## 11. SERVER TO CLIENT (FINAL STREAM FEATURES)

```
<stream:stream ...>  
  <stream:features>  
    <bind xmlns="urn:iETF:params:xml:ns:xmpp-bind" />  
    <session xmlns="urn:iETF:params:xml:ns:xmpp-session" />  
    <sm xmlns="urn:xmpp:sm:3" />  
    <csi xmlns="urn:xmpp:csi:0" />  
    <ver xmlns="urn:xmpp:features:rosterver" />  
  </stream:features>
```

# STANZAS

```
<message to='some@domain.tld' from='d@conversations.im'>  
  <body>Hi</body>  
</message>
```

```
<presence to='d@conversations.im' from='somebody@domain.tld'>  
  <show>away</show>  
</presence>
```

```
<iq to='somebody@domain.tld' from='d@conversations.im' type='get'  
  <payload xmlns='urn:example:some-namespace' />  
</iq>
```

## MESSAGE

- Do not expect a response
- Can be put into offline queue
- With or without body (human readable part)

## PRESENCE

- Status (availability) information

## IQ (INFO/QUERY)

- Expect a response
- Types
  - Request: `get / set`
  - Response: `result / error`
- Must contain `id`

# ADDRESSING

- Bare JID: `user@example.com`
- Domain JID: `example.com`
- Full JID: `user@example.com/desktop.Uf34`
- None: implicit to & from own account

# RESOURCE BINDING

## 12.CLIENT TO SERVER

```
<iq id='one' type='set'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <resource>desktop.Uf34</resource>
  </bind>
</iq>
```

## 13. SERVER TO CLIENT

```
<iq id='one' type='result'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <jid>d@conversations.im/desktop.Uf34</jid>
  </bind>
</iq>
```

# SESSIONS (WELCOME TO XMPP <sup>1</sup>)

- Part of RFC 3921 (XMPP: IM and Presence)
- Not even mentioned in RFC 6120 & RFC 6121
- Has been replaced by *initial presence*?
- Old servers still require it

<sup>1</sup> All protocols really

```
<stream:features>
  <session xmlns="urn:ietf:params:xml:ns:xmpp-session">
    <optional />
  </session>
  ...
</stream:features>
```

## 14. CLIENT TO SERVER

```
<iq to='conversations.im'
  type='set'
  id='two'>
  <session xmlns='urn:ietf:params:xml:ns:xmpp-session' />
</iq>
```

## 15. SERVER TO CLIENT

```
<iq to='d@conversations.im/desktop.Uf34' id='two' type='result'
```

# INITIAL PRESENCE

```
<presence from='d@conversations.im/desktop.Uf34'>  
  <show>chat</show>  
  <status>Hey there. I'm using Conversations!</status>  
</presence>
```

- Presence transmitted to contacts
- Contacts' presences send to client
- Start receiving messages

# SERVICE DISCOVERY (REQUESTS)

## CLIENT TO SERVER

```
<iq to='conversations.im' from='d@conversations.im/desktop.Uf34'  
  type='get' id='three'>  
  <query xmlns='http://jabber.org/protocol/disco#info' />  
</iq>
```

## CLIENT TO SERVER

```
<iq from='d@conversations.im/desktop.Uf34' type='get' id='four'>  
  <query xmlns='http://jabber.org/protocol/disco#info' />  
</iq>
```

# SERVICE DISCOVERY (RESPONSES)

## SERVER TO CLIENT

```
<iq to='d@conversations.im/desktop.Uf34' from='conversations.im'  
  type='result' id='three'>  
  <query xmlns='http://jabber.org/protocol/disco#info'>  
    <feature var='urn:xmpp:blocking' />  
  </query>  
</iq>
```

## SERVER TO CLIENT

```
<iq to='d@conversations.im/desktop.Uf34' from='d@conversations.i  
  type='result' id='four'>  
  <query xmlns='http://jabber.org/protocol/disco#info'>  
    <feature var='urn:xmpp:mam:2' />  
    <feature var='urn:xmpp:carbons:2' />  
  </query>  
</iq>
```

# ITEM DISCOVERY (REQUEST)

## CLIENT TO SERVER

```
<iq to='conversations.im' from='d@conversations.im/desktop.Uf34'  
  type='set' id='five'>  
  <query xmlns='http://jabber.org/protocol/disco#items' />  
</iq>
```

# ITEM DISCOVERY (RESPONSE)

## SERVER TO CLIENT

```
<iq to='d@conversations.im/desktop.Uf34' from='conversations.im'  
  type='result' id='five'>  
  <query xmlns='http://jabber.org/protocol/disco#items'>  
    <item jid='upload.conversations.im' name='HTTP Upload '/>  
    <item jid='muc.conversations.im' name='Group Chat' />  
  </query>  
</iq>
```

# SERVICE DISCOVERY (SUBSEQUENT REQUESTS)

## CLIENT TO SERVER

```
<iq from='d@conversations.im/desktop.Uf34'  
    to='upload.conversations.im' type='get' id='six'>  
  <query xmlns='http://jabber.org/protocol/disco#info' />  
</iq>
```

## CLIENT TO SERVER

```
<iq from='d@conversations.im/desktop.Uf34'  
    to='muc.conversations.im' type='get' id='seven'>  
  <query xmlns='http://jabber.org/protocol/disco#info' />  
</iq>
```

# SERVICE DISCOVERY (SUBSEQUENT RESPONSES)

## SERVER TO CLIENT

```
<iq to='d@conversations.im/desktop.Uf34'  
  from='upload.conversations.im' type='result' id='six'>  
  <query xmlns='http://jabber.org/protocol/disco#info'>  
    <feature var='urn:xmpp:http:upload:0' />  
  </query>  
</iq>
```

## SERVER TO CLIENT

```
<iq to='d@conversations.im/desktop.Uf34'  
  from='muc.conversations.im' type='result' id='seven'>  
  <query xmlns='http://jabber.org/protocol/disco#info'>  
    <feature var='http://jabber.org/protocol/muc' />  
  </query>  
</iq>
```

# SERVICE DISCOVERY (ME TOO)

## CLIENT TO SERVER

```
<iq to='alice@example.com/mobile.Xab3'  
  from='d@conversations.im/desktop.Uf34'  
  type='result' id='x'>  
  <query xmlns='http://jabber.org/protocol/disco#info'>  
    <feature var='urn:xmpp:jingle:1' />  
    <feature var='urn:xmpp:jingle:apps:file-transfer:5' />  
    <feature var='urn:xmpp:jingle:transports:s5b:1' />  
    <feature var='urn:xmpp:jingle:transports:ibb:1' />  
  </query>  
</iq>
```

# FEATURES

- Roster + Roster versioning (stream feature)
- Stream Management (stream feature)
- Client State Indication (stream feature)
- Blocking Command (server disco)
- Message Archive Management (account disco)
- Message Carbons (server disco???)

# ROSTER + BLOCKING COMMAND

- IQ (type=get) to get a list of Jabber IDs
- IQ (type=set) to modify
- IQ (type=set) from account to client for pushes

```
<iq type='get' id='random'>  
  <blocklist xmlns='urn:xmpp:blocking' />  
</iq>
```

# STREAM MANAGEMENT

## CLIENT TO SERVER

```
<enable xmlns='urn:xmpp:sm:3' />

<message to='alice'>
  <body>Hi</body>
</message>
<r xmlns='urn:xmpp:sm:3' />

<a h='1' xmlns='urn:xmpp:sm:3' />
```

## SERVER TO CLIENT

```
<enabled xmlns='urn:xmpp:sm:3'
  id='random-id' resume='true' />

<a xmlns='urn:xmpp:sm:3' h='1' />
<message from='alice' />
<r xmlns='urn:xmpp:sm' />
```

# STREAM MANAGMENT - RESUME

```
<resume h='1' previd='random-id' xmlns='urn:xmpp:sm:3' />
```

# MESSAGE CARBONS - WHY?

- Traditionally:
  - Message to full Jabber IDs *or*
  - Message to bare JID goes to **most available**
- Carbons:
  - Receive *otherwise not received* messages as copy
  - Receive outgoing as copy

# MESSAGE CARBONS (COPIES)

```
<message to='d@conversations.im/desktop.Uf34'>
  <received xmlns='urn:xmpp:carbons:2'>
    <forwarded xmlns='urn:xmpp:forward:0'>
      <message xmlns='jabber:client'
        from='alice@conversations.im/mobile.Xab3'
        to='d@conversations.im/mobile.H3x0'
        type='chat'>
        <body>Hi. What's up?</body>
      </message>
    </forwarded>
  </received>
</message>
```

# MESSAGE CARBONS (ENABLE)

```
<iq from='d@conversations.im/desktop.Uf34' type='set' id='eleven'  
  <enable xmlns='urn:xmpp:carbons:2' />  
</iq>
```

# MAM (LIVE MESSAGES)

```
<message from='alice@conversations.im/mobile.Xab3'  
  to='d@conversations.im/desktop.Uf34'>  
  <body>Boring sample text</body>  
  <stanza-id xmlns='urn:xmpp:sid:0' by='d@conversations.im'  
    id='1568719955' />  
</message>
```

# MAM (CATCHUP)

```
<iq type='set' id='thirteen'>
  <query xmlns='urn:xmpp:mam:2' queryid='ran'>
    <set xmlns='http://jabber.org/protocol/rsm'>
      <after>1568719955</after>
    </set>
  </query>
</iq>
```

# MAM (FAN OUT)

```
<message to='d@conversations.im/desktop.Uf34'>
  <result xmlns='urn:xmpp:mam:2' queryid='ran' id='1568724977'>
    <forwarded xmlns='urn:xmpp:forward:0'>
      <delay xmlns='urn:xmpp:delay'
        stamp='2019-09-17T12:56:17Z' />
      <message xmlns='jabber:client'
        from='alice@conversations.im/mobile.Xab3'
        to='d@conversations.im'
        type='chat'>
        <body>Hi. What's up?</body>
      </message>
    </forwarded>
  </result>
</message>
```

# MAM (IQ RESULT / FIN)

```
<iq type='result' id='thirteen'  
  to='d@conversations.im/desktop.Uf34'>  
  <fin xmlns='urn:xmpp:mam:2'>  
    <set xmlns='http://jabber.org/protocol/rsm'>  
      <first>1568724977</first>  
      <last>1568726327</last>  
    </set>  
  </fin>  
</iq>
```

# RECAP

1. STARTTLS
2. Login
3. Bind<sup>1</sup>
4. Stream Management
5. Service Discovery
6. Message Carbons
7. Initial Presence
8. MAM Catchup<sup>2</sup>

<sup>1</sup> followed by <session/> on old servers

<sup>2</sup> last id gathered **before** initial presence

# RESOURCES

- RFC 6120 + RCF 6121
- [XEP-0412: XMPP Compliance Suites 2019](#)
- Gajim's XML console
- [jdev@muc.xmpp.org](mailto:jdev@muc.xmpp.org)