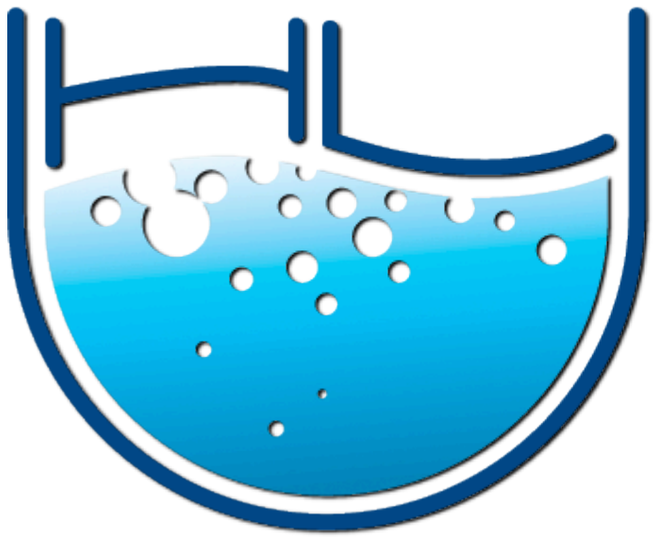
Adventures In XMPP

Kevin A. Smith



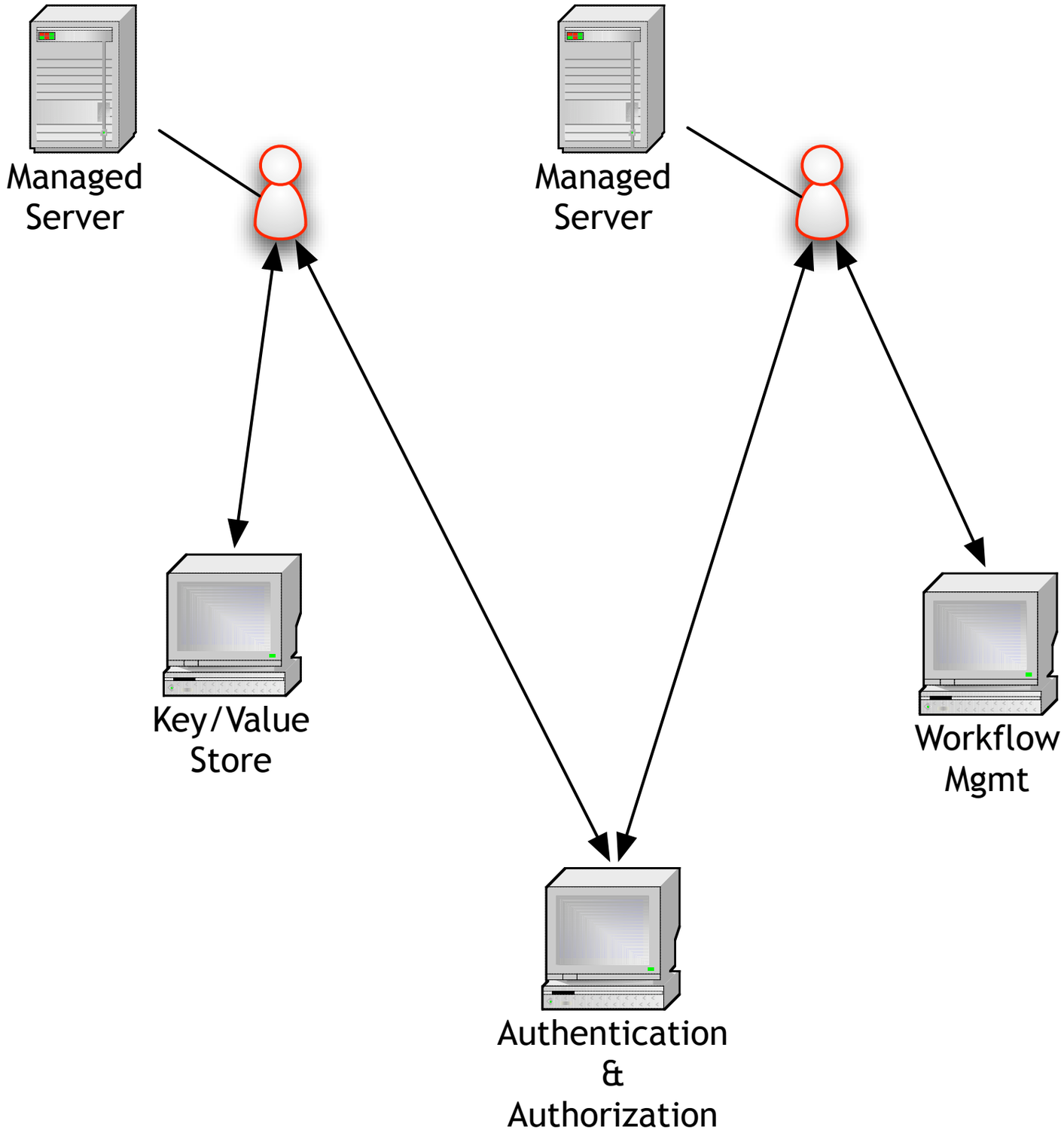
Hypothetical Labs

Hypothetical Labs

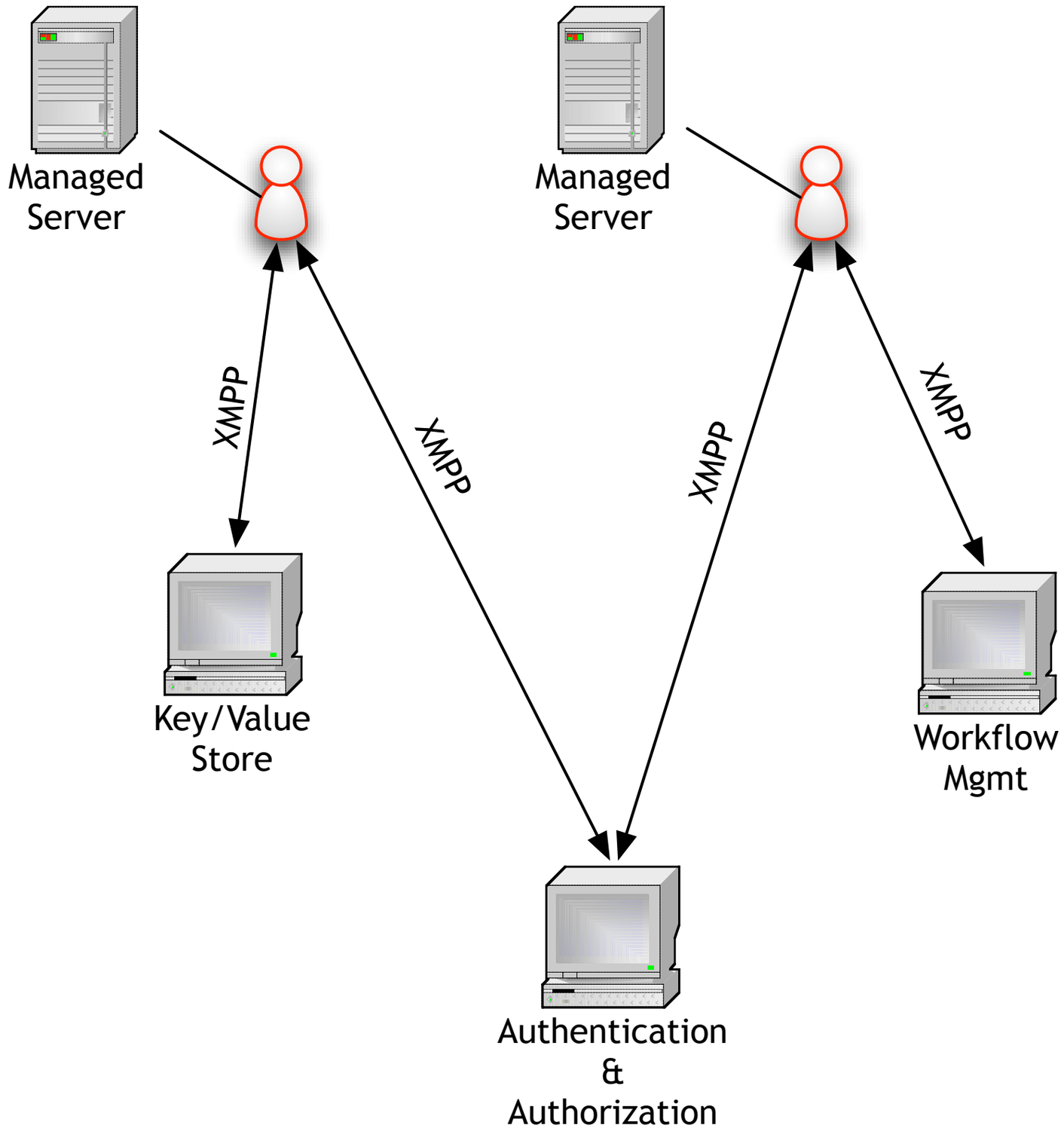
Vertebra

Hypothetical Labs


R.I.P.



Hypothetical Labs



Hypothetical Labs

What Are IQ Packets?

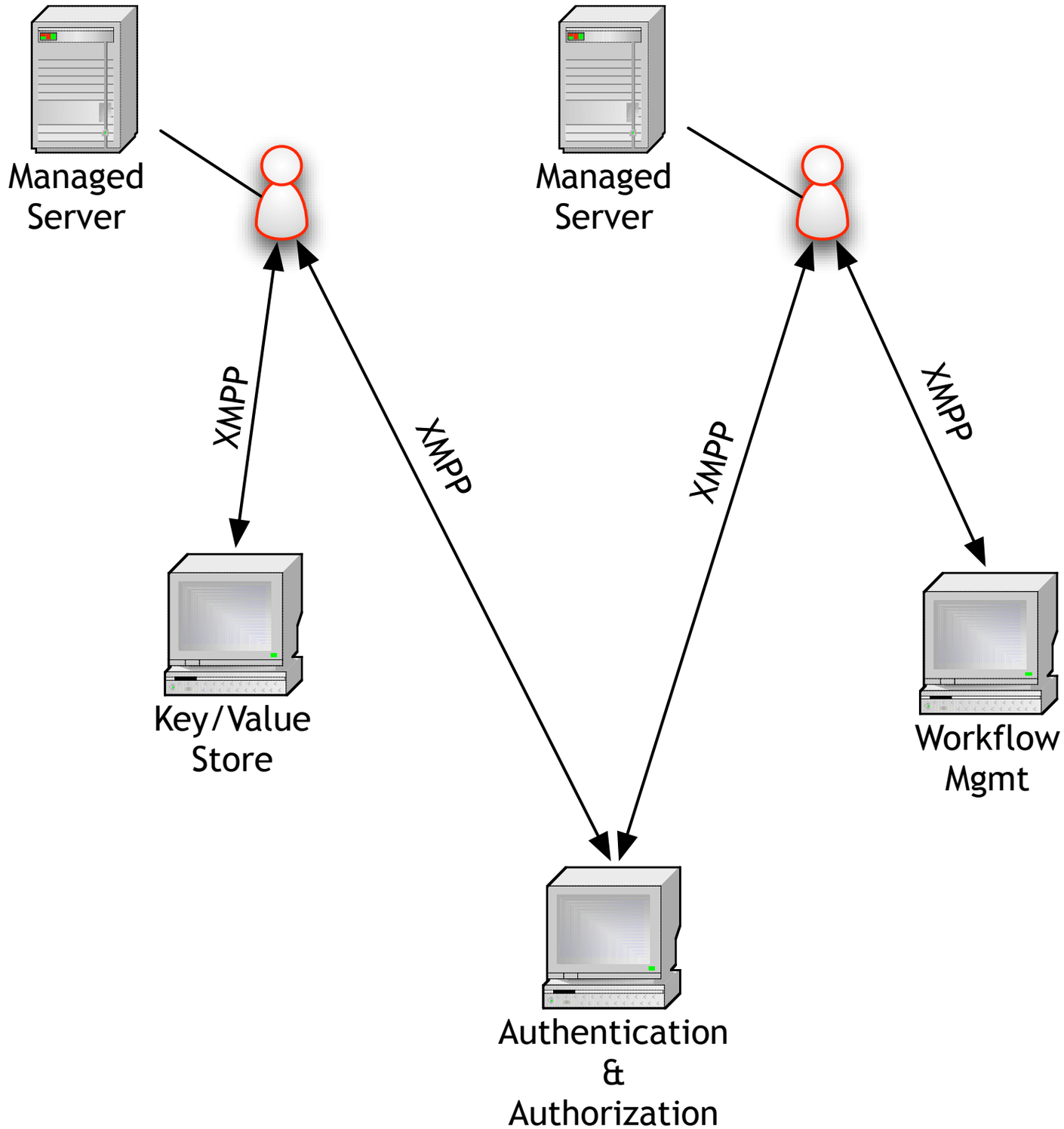
- Models request/response "conversations"
- Packet ids unique to an exchange
- Packet types: set, get, result, error

Hypothetical Labs

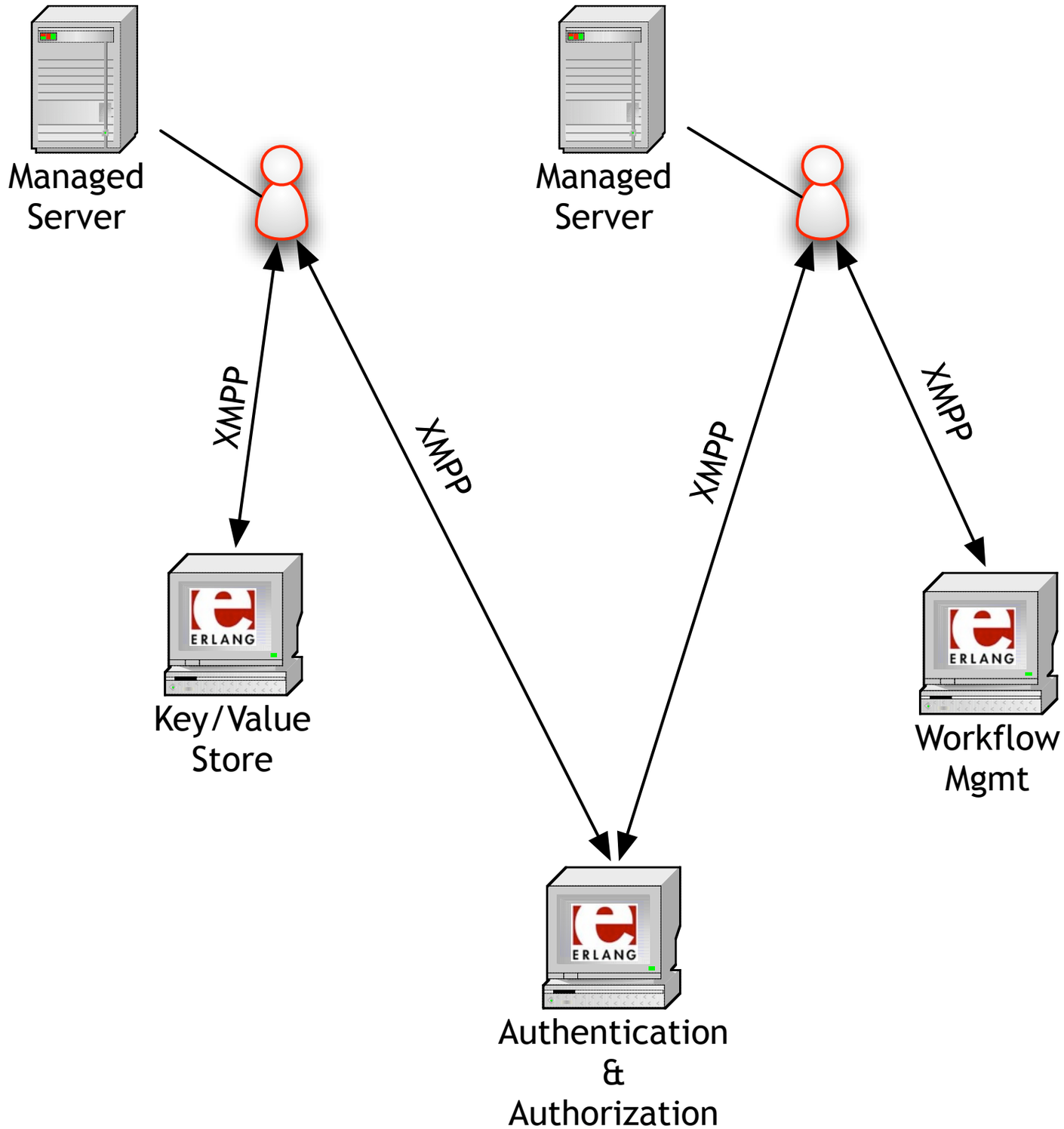
IQ Packet

```
<iq id=”123” type=”set”
    from=”foo@localhost” to=”bar@localhost”>
  <event id=”98432” name=”Erlang Factory” />
</iq>
```

Hypothetical Labs



Hypothetical Labs



Hypothetical Labs

What Are My Options?

1. ejabberd module

Hypothetical Labs

Pros

- Fast
- Mostly easy
- Obvious

Hypothetical Labs

Cons

- Wholly dependent on ejabberd
- Complicated deployment
- Lacking ejabberd docs

Hypothetical Labs

2. Jabberlang

Hypothetical Labs

Pros

- It exists
- (Mostly) works
- Removes ejabberd dependency

Hypothetical Labs

Cons

- (Mostly) works
- Blocking send/receive
- Orphaned

Hypothetical Labs

3. exmpp

4. Write our own

Hypothetical Labs

Pros

- Complete control over implementation
- Implement only what we need
- Tailored to integration needs

Hypothetical Labs

Cons

- Complete control over implementation
- Never written an XMPP client before
- Tight schedule

Hypothetical Labs

Introducing natter

What It Is

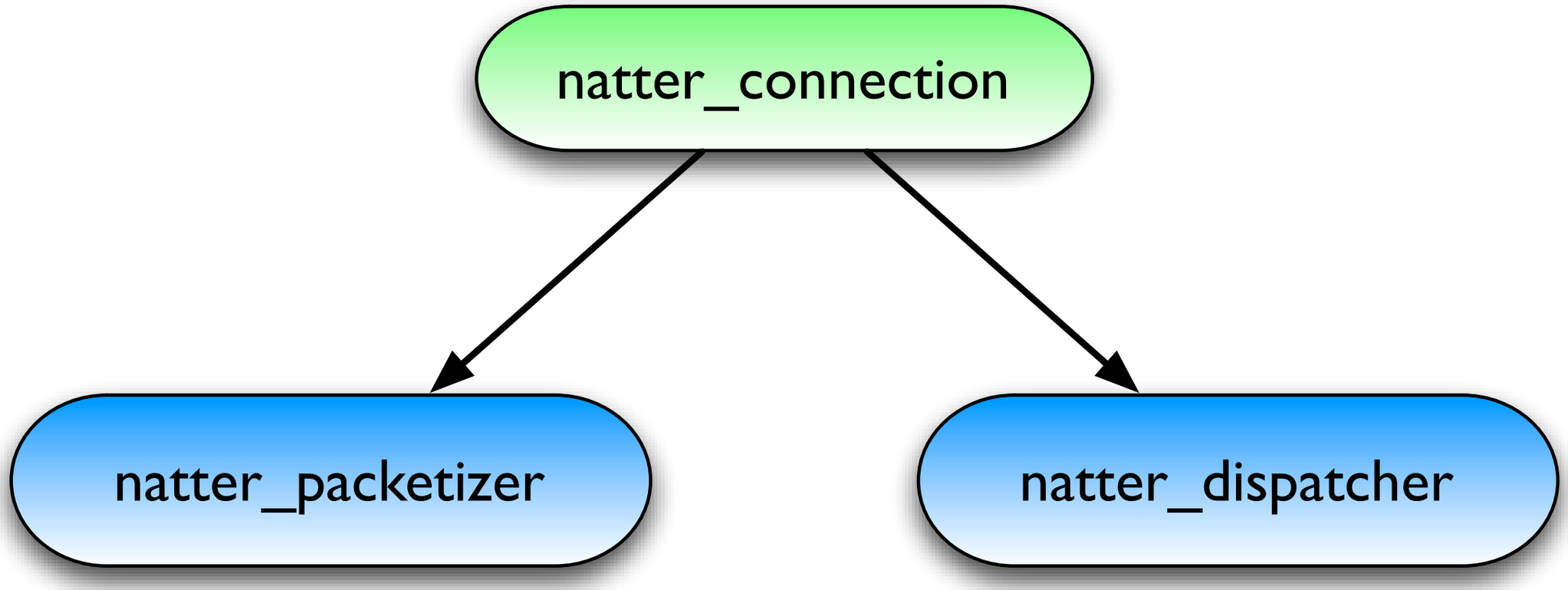
- XMPP library
- Computer-to-computer via XMPP
- **IQ only**

Hypothetical Labs

What It Isn't

- General purpose chat library
- No message support
- No rosters
- Minimal presence

Hypothetical Labs



Hypothetical Labs

Connecting

```
Config = [{host, “localhost”},
          {user, “foo”},
          {password, “bar”},
          {resource, “foo”}],
{ok, Cn} = natter_connection:start_link(Config).
```

Hypothetical Labs

Receiving XMPP Messages

Exchanges

- Routes packets to interested processes
- XMPP messages are async

Hypothetical Labs

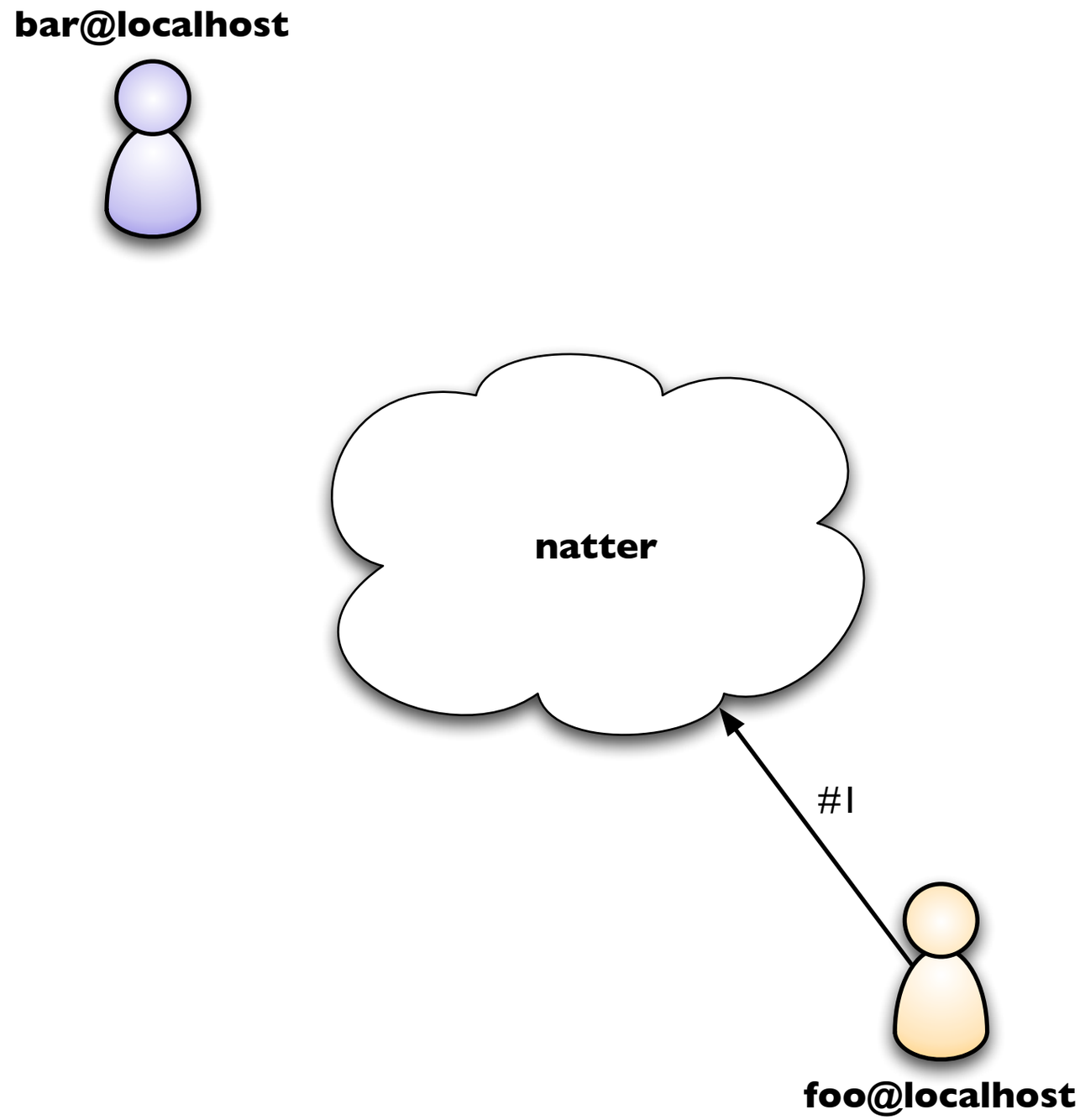
```
{ok, Cn} = natter_connection:start_link(Config),

natter_connection:register_default_exchange(self(), Cn).
```

```
natter_connection:register_exchange(Cn,
                                    “iq”,
                                    “bar@localhost”,
                                    self()).
```

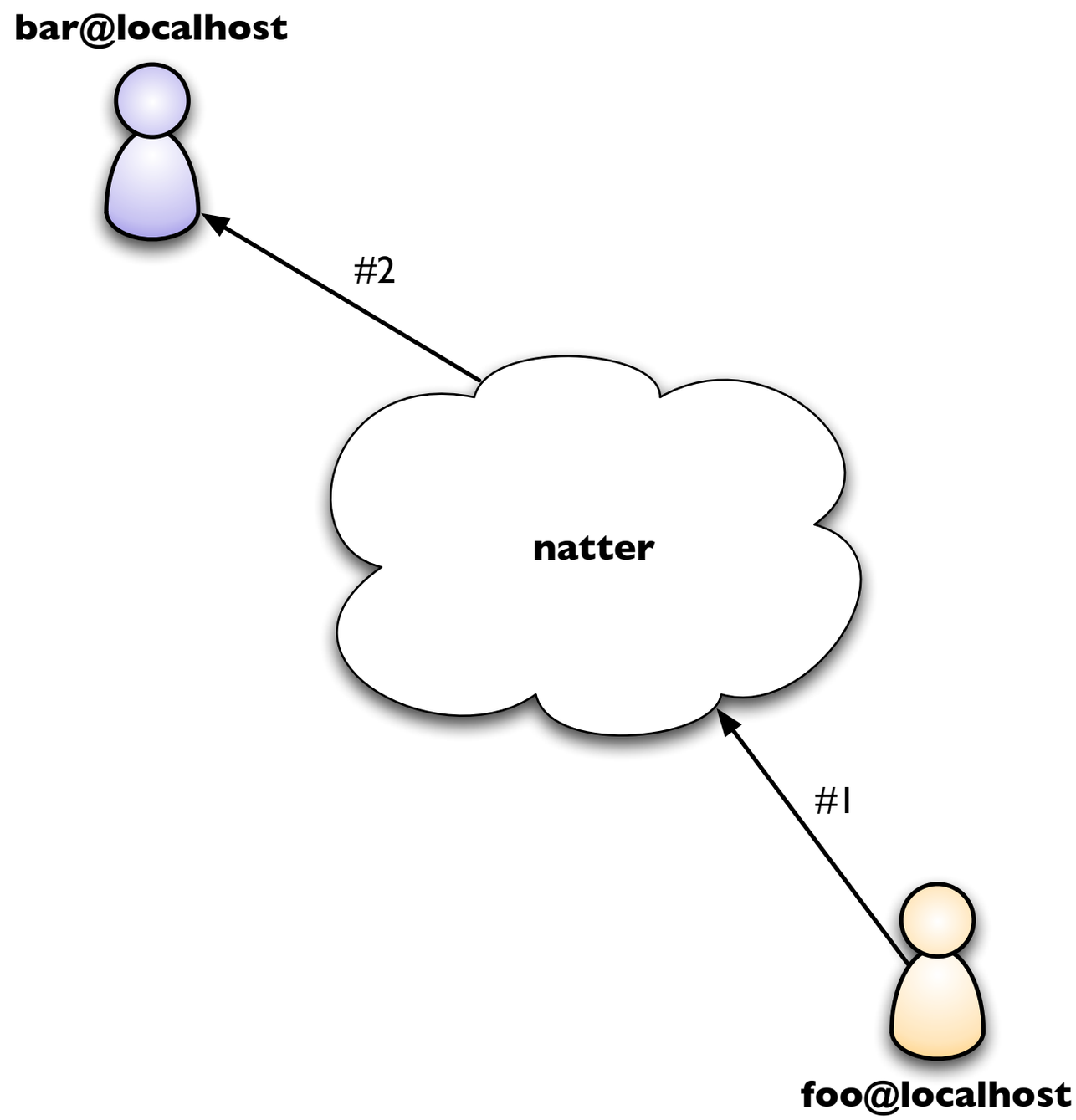
Temporary Exchanges

- Used to route incoming responses
- Live for a single IQ “conversation”

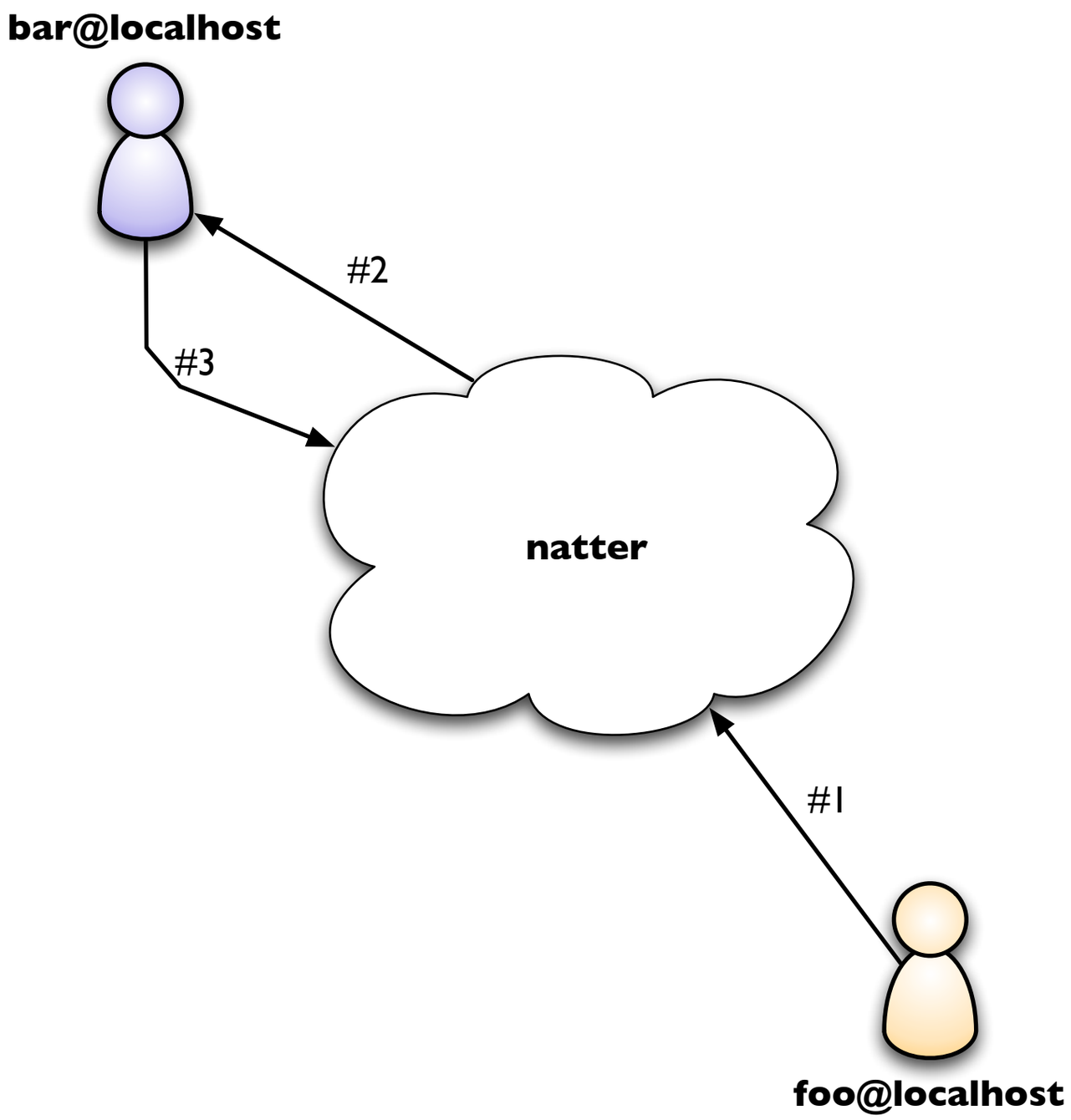
Hypothetical Labs



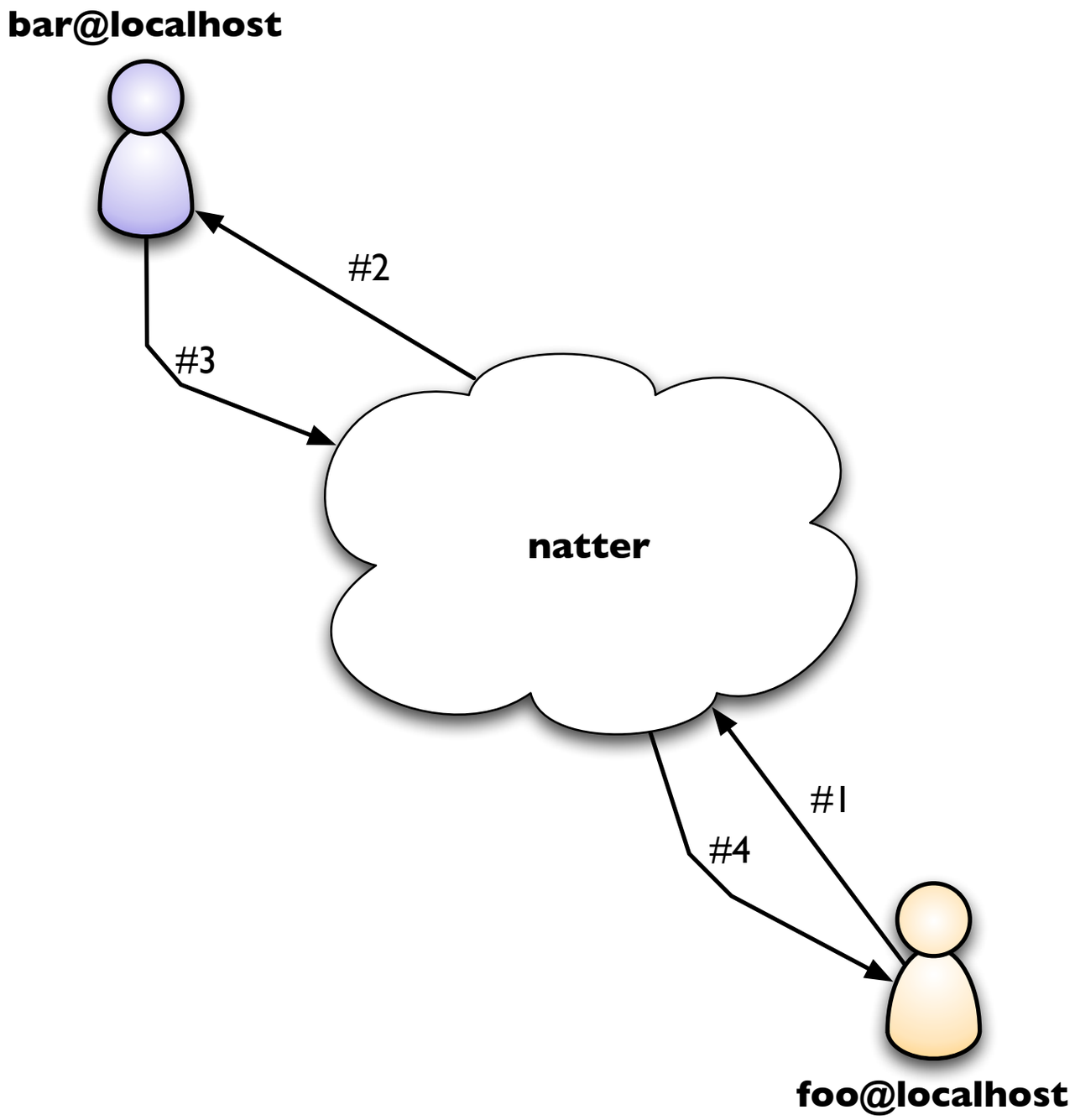
Hypothetical Labs



Hypothetical Labs



Hypothetical Labs



Hypothetical Labs

XML Parsing

- Small C wrapper around libexpat
- Inspired by Jabberlang
- Faster than xmerl

Hypothetical Labs

Parsed XMPP

```
{xmlelement, “iq”, Attrs, Subels}
```

Delivering XMPP

- Sent as an Erlang message
- `handle_info()` or `receive` block

Hypothetical Labs

```
{packet, {xmlelement, “iq”, [{id, “101”}],
          [{xmlelement, “hello-world”, [], []}]}}
```

Sending XMPP Messages

Nonblocking (Fire and forget)

Hypothetical Labs

```
Payload = {xmlelement, “hello-world”, [], []},
natter_connection:send_iq(Cn, “result”, “100”,
                          “foo@localhost”, Payload)
```

Blocking (Request & Response)

```
Payload = {xmlelement, “hello-world”, [], []},
natter_connection:send_wait_iq(Cn, “set”, “100”,
                               “foo@localhost”,
Payload).
```

Other Features

- Pluggable fuzzing engine
- Reconnect and recovery
- Duplicate suppression

Hypothetical Labs

Things I Learned

Hypothetical Labs

Know What You're Building Before You Build It

When Do You Test?

When Do You Test?

All the f**king time!

Hypothetical Labs

Dialyzer is your friend

So are typespecs

Processes are just like objects

Functions should be short

Modules should do one thing

Hands-On Erlang
7/15 - 7/16
Chicago, IL

http://handsonerlangchicago.eventbrite.com