Alexandre Lachèze
Pierre Guilleminot

KadOH

Kademlia over HTTP

a Javascript framework bringing DHT to mobile applications

What have been done so far in the DHT* landscape ?

* Distributed Hash Table

Theoretical resuts

Chord

Kademlia

Improvement proposals

Public large scale implementations

File sharing

eMule

BitTorrent’s Mainline

What is missing ?

Mobiles

Ignored by DHTs

Huge and growing audience

Private data

Web Applications

taking advantage of DHTs scalability

Beyond file sharing...

overlay networks

improving privacy protections

What have we done ?

A Javascript framework implementing Kademlia in browsers using XMPP over HTTP

a.k.a. **KadOH**

Why Javascript ?

- improved portability vs. native
- good performance
- event-driven asynchronous I/O
- simplified distribution process
- automated backward compatibility

Why XMPP ?

How to establish peer to peer connections ?

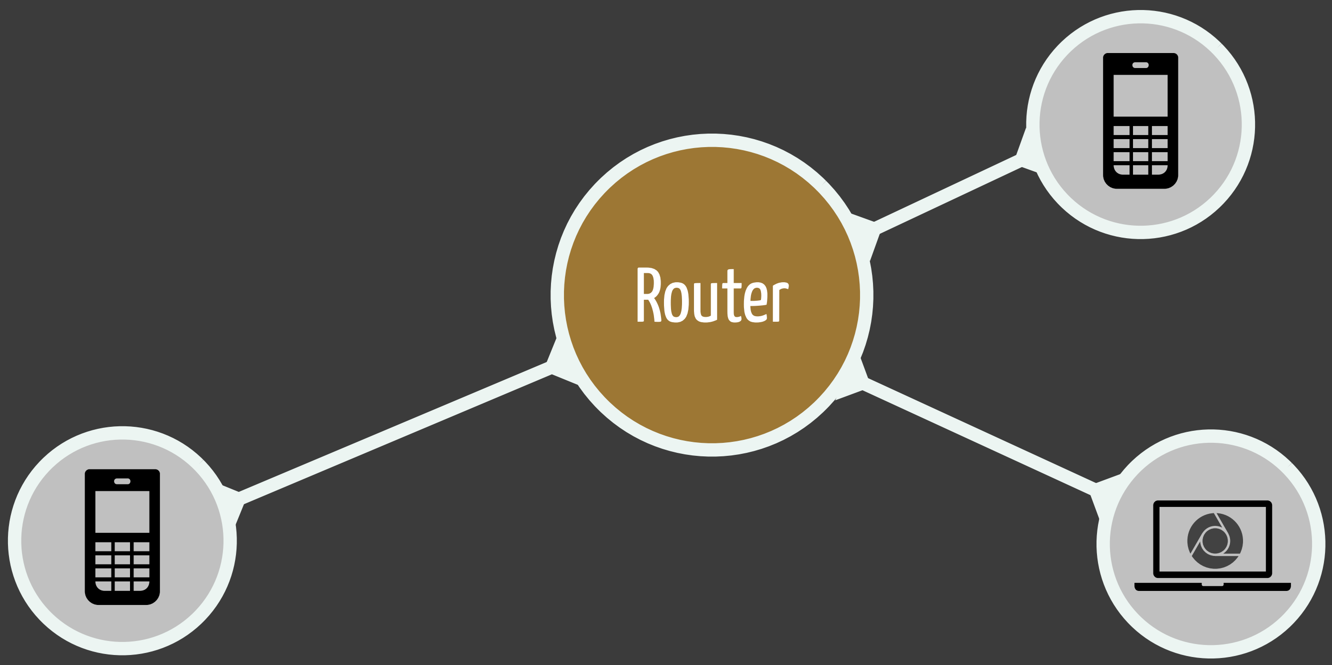
Common problem on obstrusive cellular networks

highly restrictive firewalls

technicaly impossible

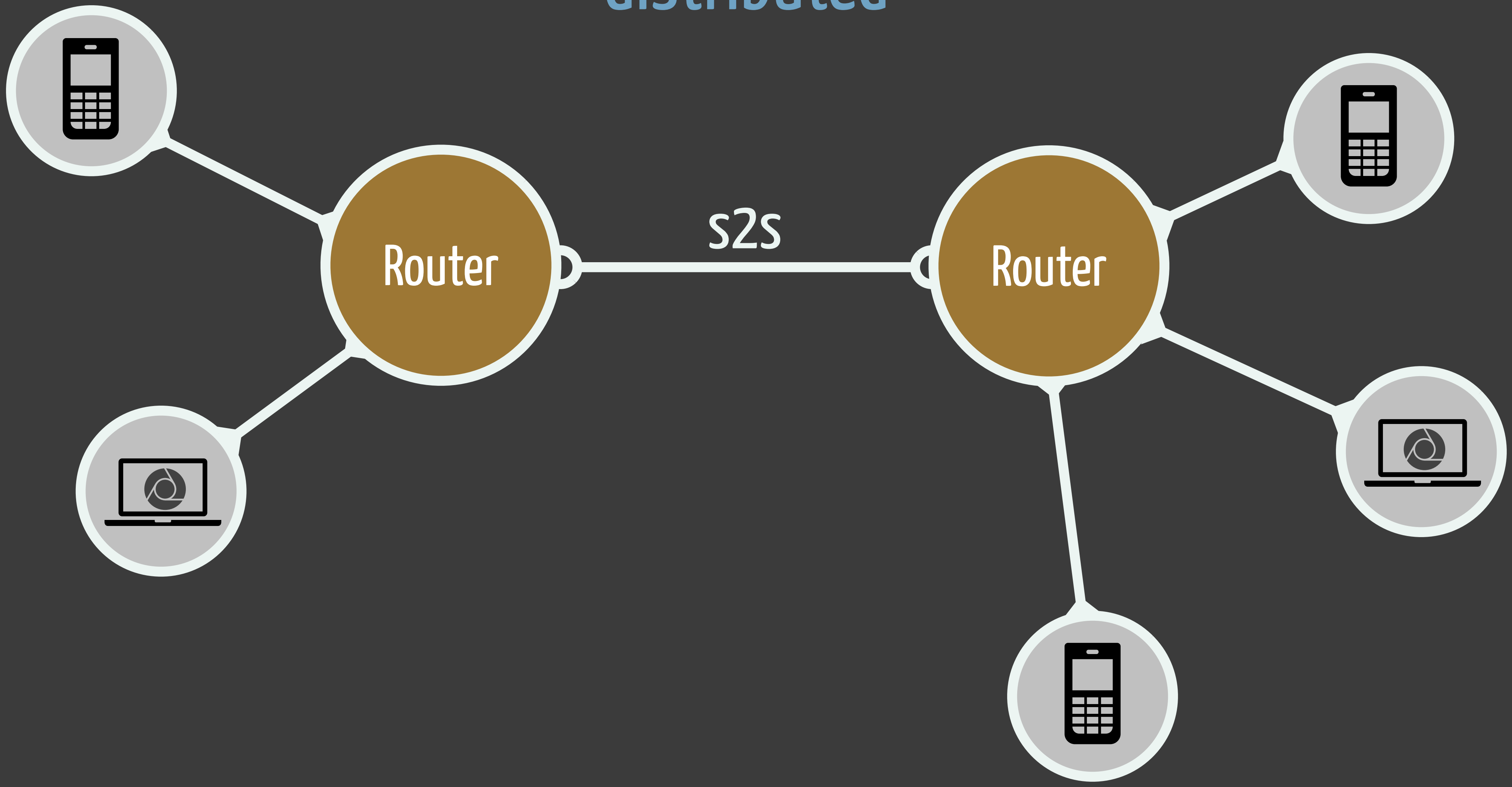
need a for third party router

Develop our own router ?



Develop our own routers ?

distributed



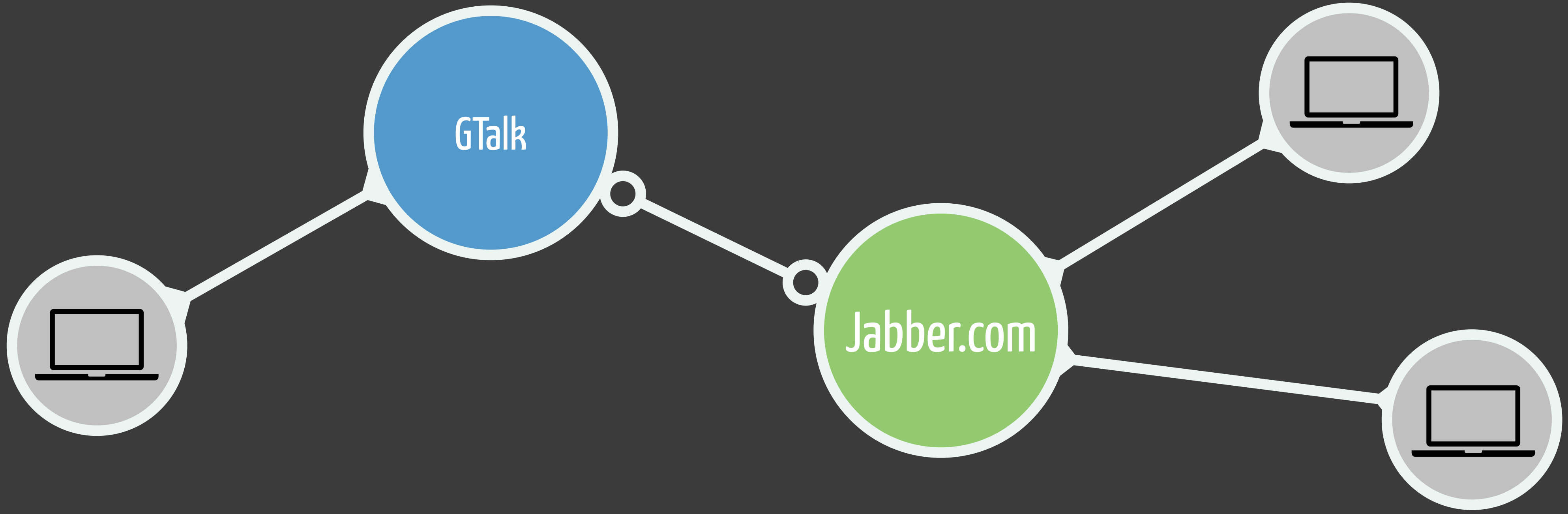
Develop our own routers ?

heavy

does not scale

against the end-to-end principle

Unsing the XMPP infrastructures



peer to peer messaging

distributed – already scaled

open

XMPP protocol

eXtensible Messaging and Presence Protocol

open and standard

based on XML

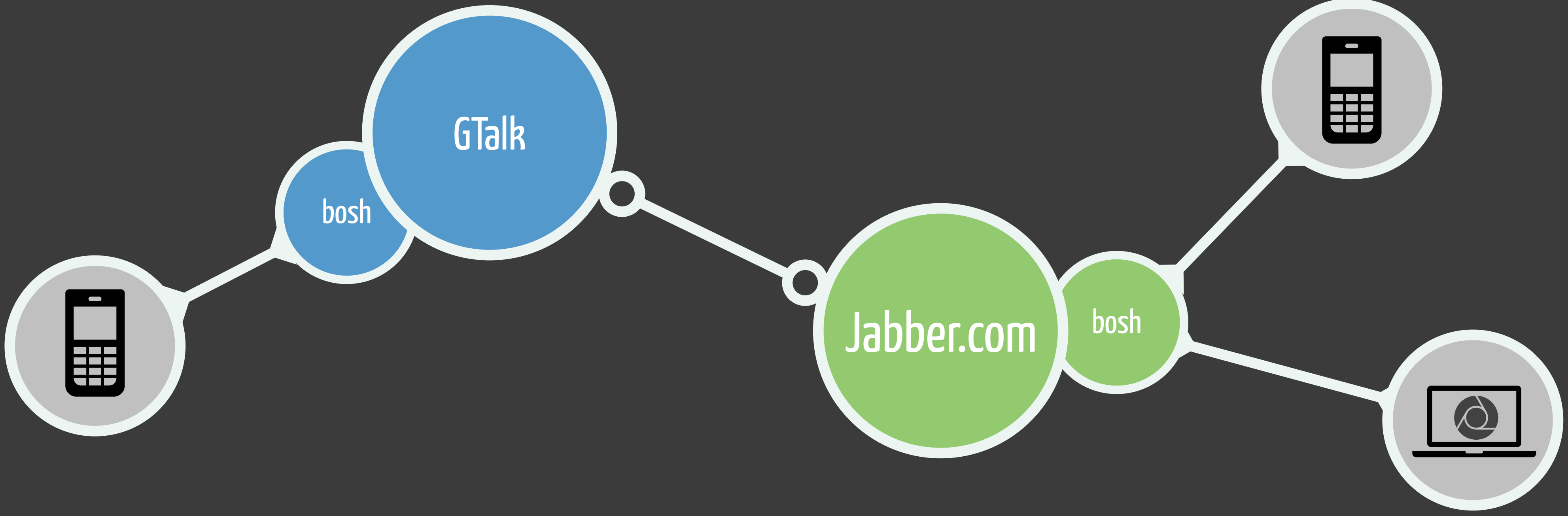
made for instant messaging

and extensible

TCP

```
<message from='juliet@jabber.org’
         to='romeo@jabber.org'
         xml:lang='en'>
  <body>Art thou not Romeo,
        and a Montague? &lt;3</body>
</message>
```

But in the browser ?



Bidirectional-streams Over Synchronous HTTP

XHR long polling

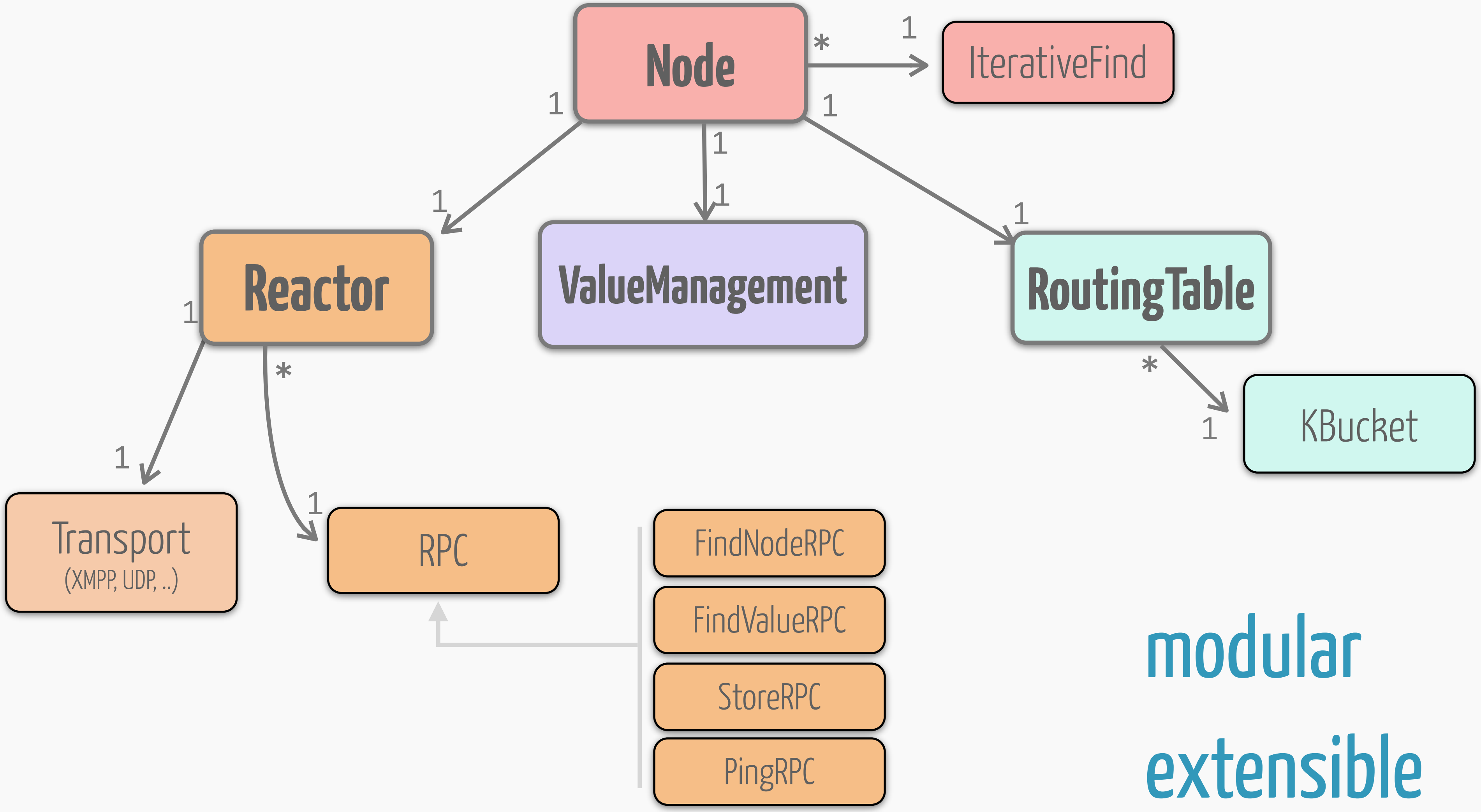
XMPP in browser

Good news :

It works !

How does it work ?

Architecture



modular

extensible

independant parts

highly hackable

How to use it ?

Instantation

```
node = new KadOH.Node(id, options)
```

Connection and joining

```
node.connect(function() {
  node.join();
});
```

Store

```
node.put('0beec7b5ea3f0fdbc95d0dd47f3c5bc275da8a33',
         'Hello world !');
```

Retrieve

```
node.get('62cdb7020ff920e5aa642c3d4066950dd1f01f4d',
         function(value) {
           alert(value);
         });
```

That’s it !

…but possibility to access to lower level functions
or totally rewrite the logic

Some extra sugar ?

- runs in **Node.js**
- supports various transports :
  - **UDP**
  - **raw XMPP**
  - **WebSocket**
- connects to the Mainline DHT

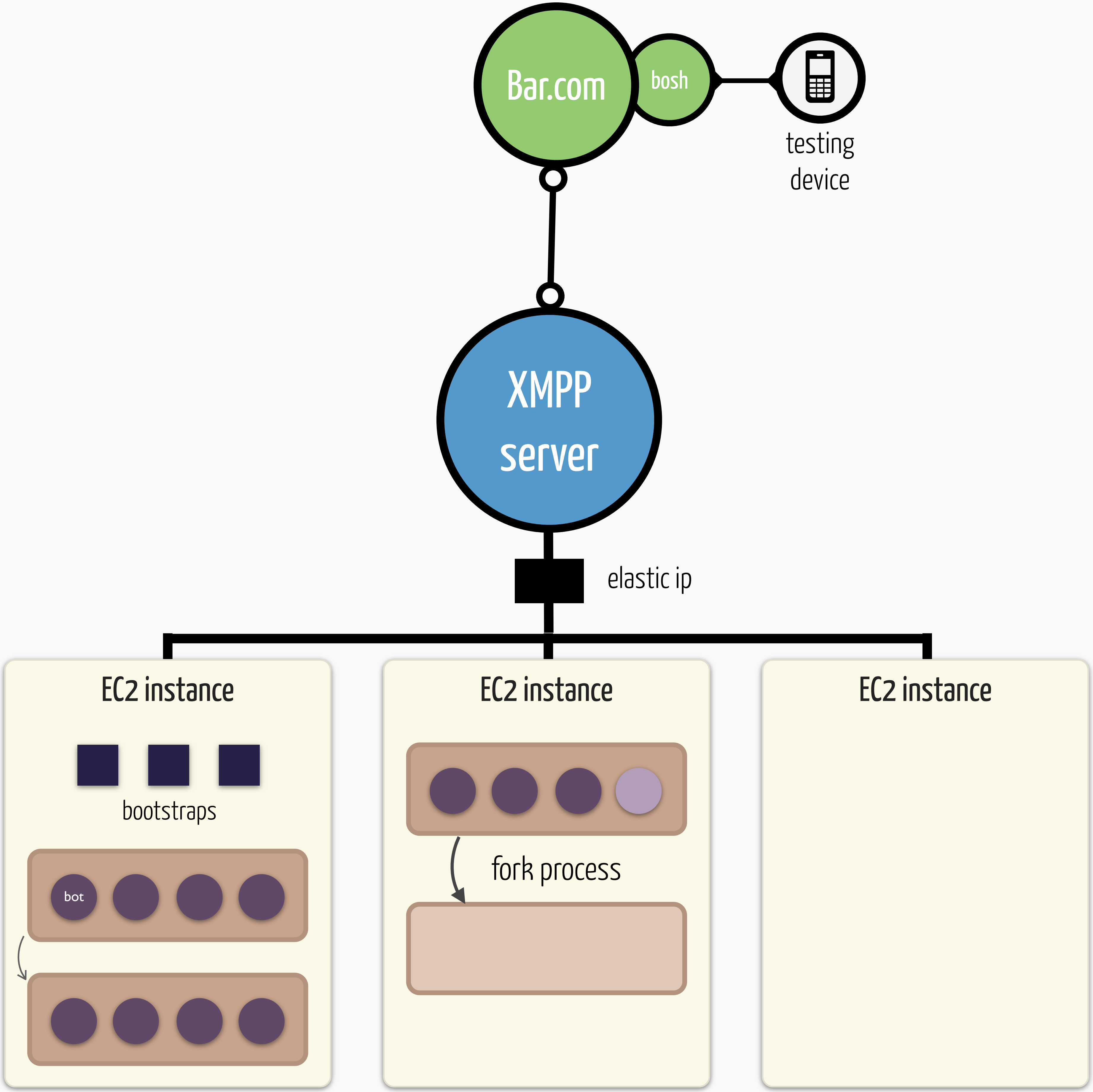
DEMO

Proxy to Mainline

How to test a DHT from scratch ?

- instantiate a large number of peers
- automatic behavior – bots
- heavy computing capacity

Amazon EC2


Bar.com
bosh
testing device
XMPP server
elastic ip
EC2 instance
EC2 instance
EC2 instance
bootstraps
bot
fork process

DEMO

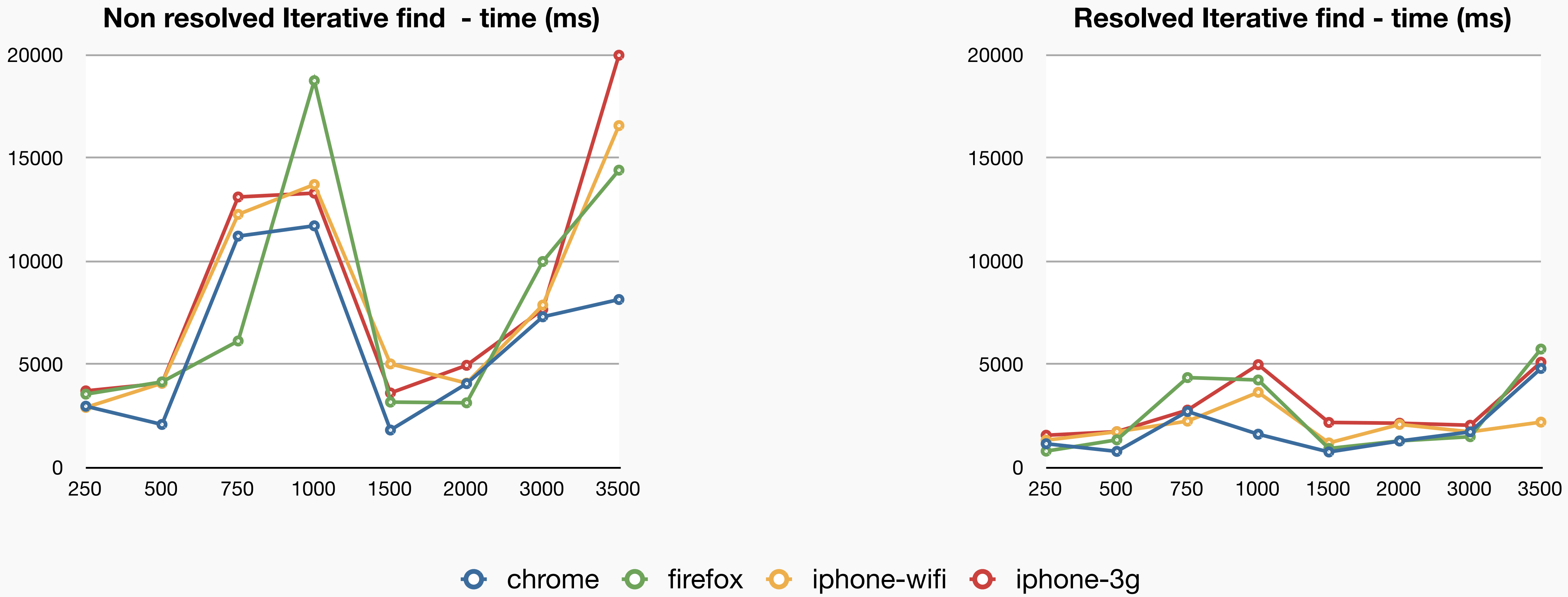
XMPP &

DHT simulation

Benchmarking

- sequence of iterative look ups
  - 5 non reaching
  - 5 reaching
- collected relevant metrics on multiple devices

Benchmarking



Non resolved Iterative find - time (ms)

Resolved Iterative find - time (ms)

Let’s sum up.

A Javascript framework implementing Kademlia in browsers using XMPP over HTTP

📱 💻

< 5s (even in3G)

KadOH

https://github.com/jinroh/kadoh

http://jinroh.github.com/kadoh/

And now ?

- still lots of features to implement
- focus on new promising technologies