

The Web Browser as Distributed Application Server: Towards Decentralized Web Applications in the Edge

Kristof Jannes, Bert Lagaisse, Wouter Joosen

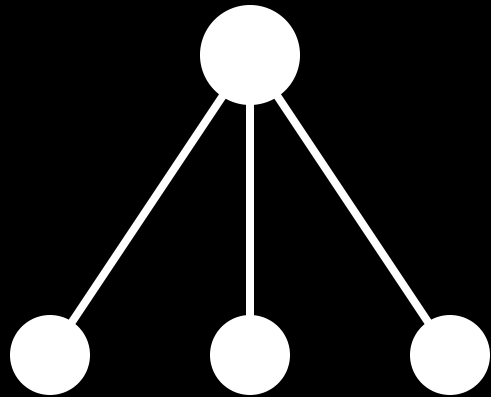
DistrINet

KU LEUVEN

Evolution of web applications

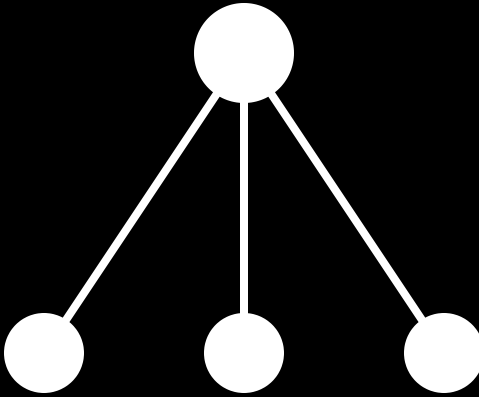
Evolution of web applications

client-server

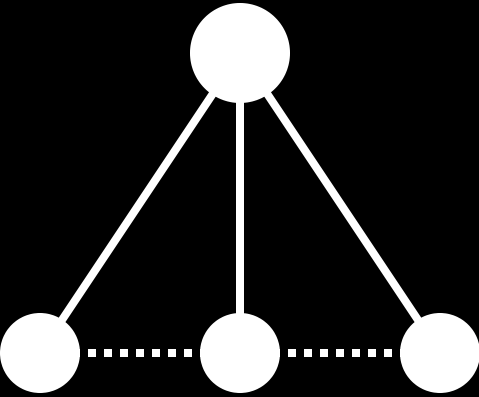


Evolution of web applications

client-server

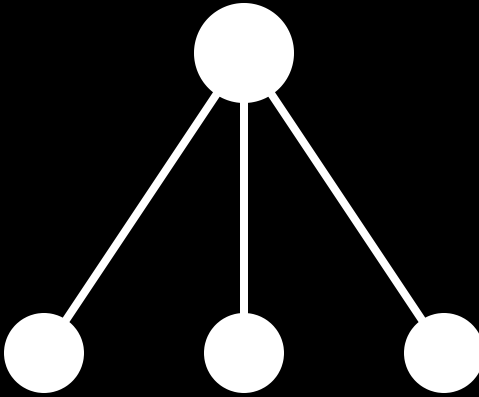


peer-to-peer

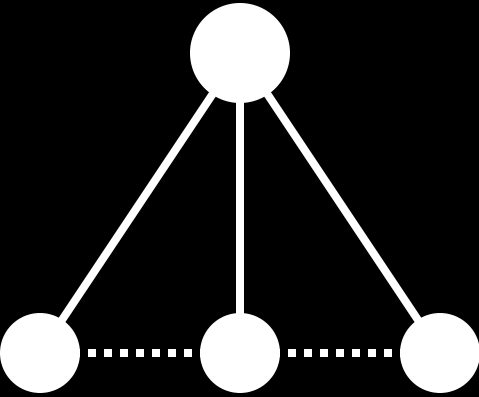


Evolution of web applications

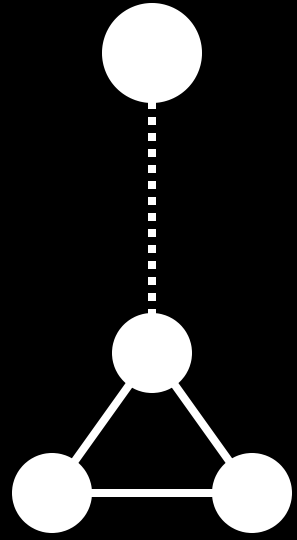
client-server



peer-to-peer

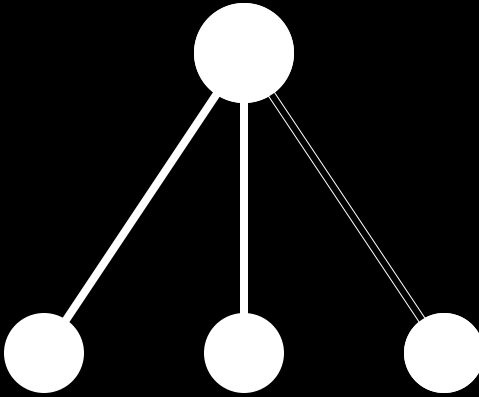


decentralized

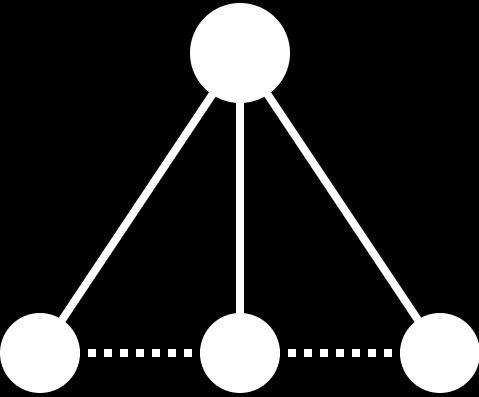


Evolution of web applications

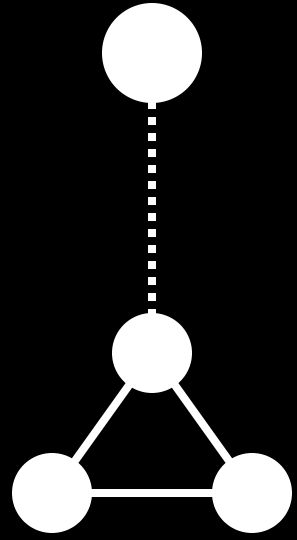
client-server



peer-to-peer

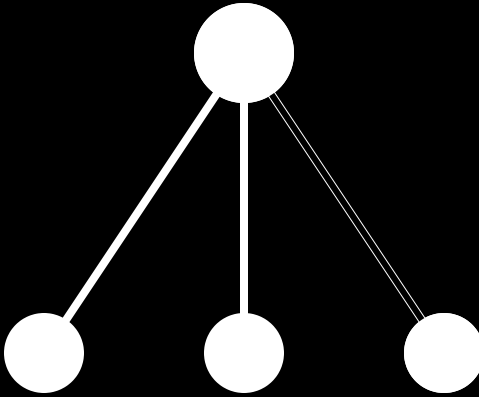


decentralized

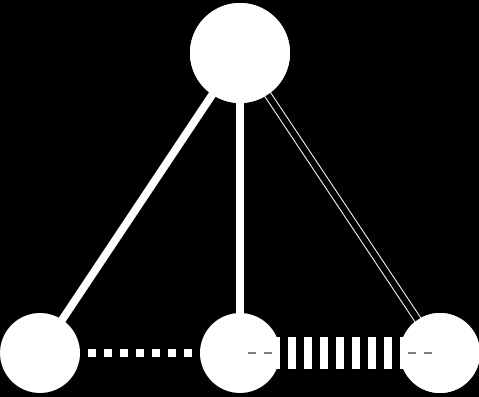


Evolution of web applications

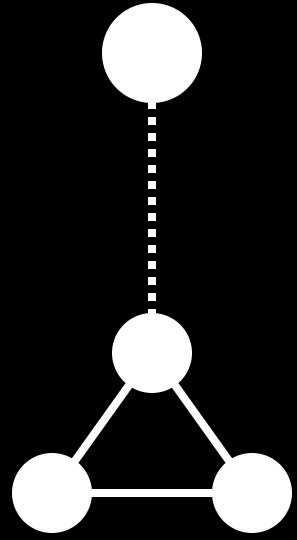
client-server



peer-to-peer

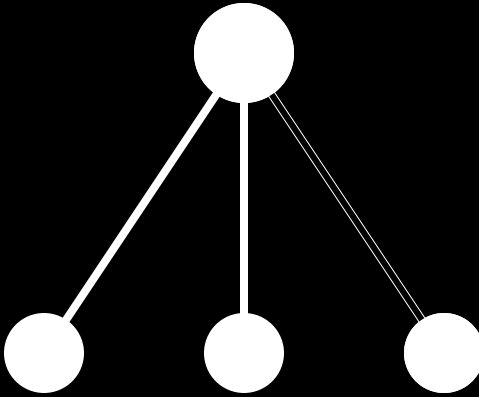


decentralized

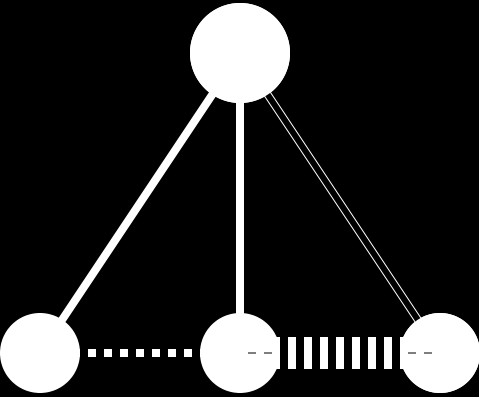


Evolution of web applications

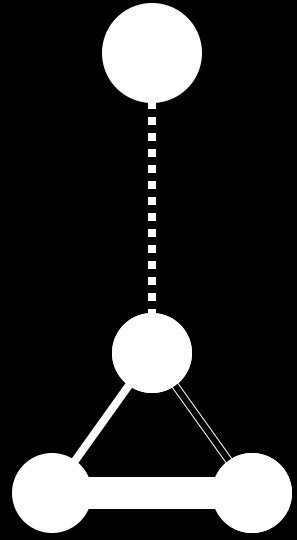
client-server



peer-to-peer

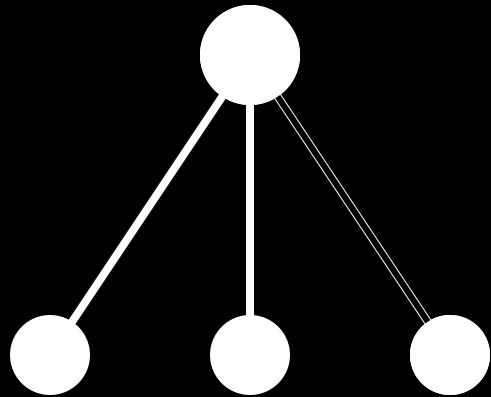


decentralized

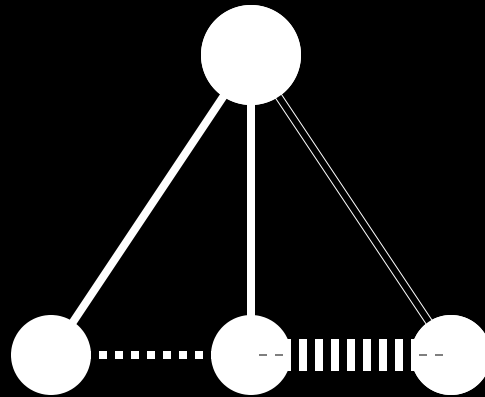


Evolution of web applications

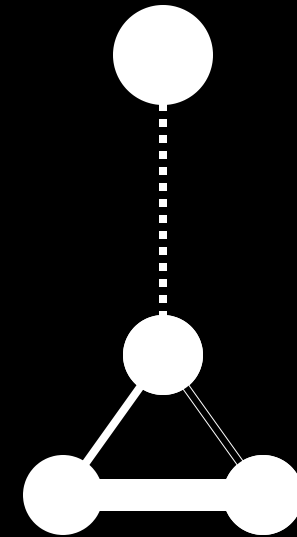
client-server



peer-to-peer



decentralized



*"Over the years, we've lost control
of our personal data"*

– Tim Berners-Lee

What do we need?

Scalability

thousands of people

Continuous operation

even when offline

You in control of your data

not the service provider

Client-centric and data-centric web middleware for decentralized web applications

Data storage, replication and synchronization

Data sharing and access control

Secure and privacy-aware data queries and analysis

The web browser is the application server
of the future

The Web Browser as Distributed Application Server

Motivational case studies

What are the requirements?

Is the browser ready?

Related work

Motivational case studies

eWhiteboard: shared whiteboard

Local P2P synchronization on LAN

Confidential data between participants

Share with colleagues and company server

eDesigners: multi-tenant graphical editing

Shared base layers for customers

Customizations by the customer

Concurrent and offline edits

eWorkforce: task planning and reporting





Technicians: often offline in the field

Back-office: mostly online but LAN network as fallback




Synchronization and conflict resolution

What are the requirements?

Basic requirements which are supported today

-  Disconnected operation
-  P2P synchronization
-  Efficient data synchronization
-  Interactive timings

Advanced requirements which are not supported today

-  User-centric access control
-  Distributed queries
-  Decentralized processing and analysis

Is the browser ready?

Is the browser ready?

Peer-to-peer communication

WebRTC

Multi-threaded execution

WebWorkers

Offline operation

ServiceWorkers

Security model

Same-Origin Policy

Is the browser ready in practice?

The web browser was not made to become an application server

Originally made for client-server requests

Not optimized for many concurrent connections

Multi-threaded operation is not straightforward

Client-centric web middleware using CRDTs

Client-centric web middleware using CRDTs

browser

Web browser

Client-centric web middleware using CRDTs

browser



Client-centric web middleware using CRDTs

application

eDrawings

eDesigners

eWorkforce

browser

Web browser



Client-centric web middleware using CRDTs

application

eDrawings

eDesigners

eWorkforce

middleware

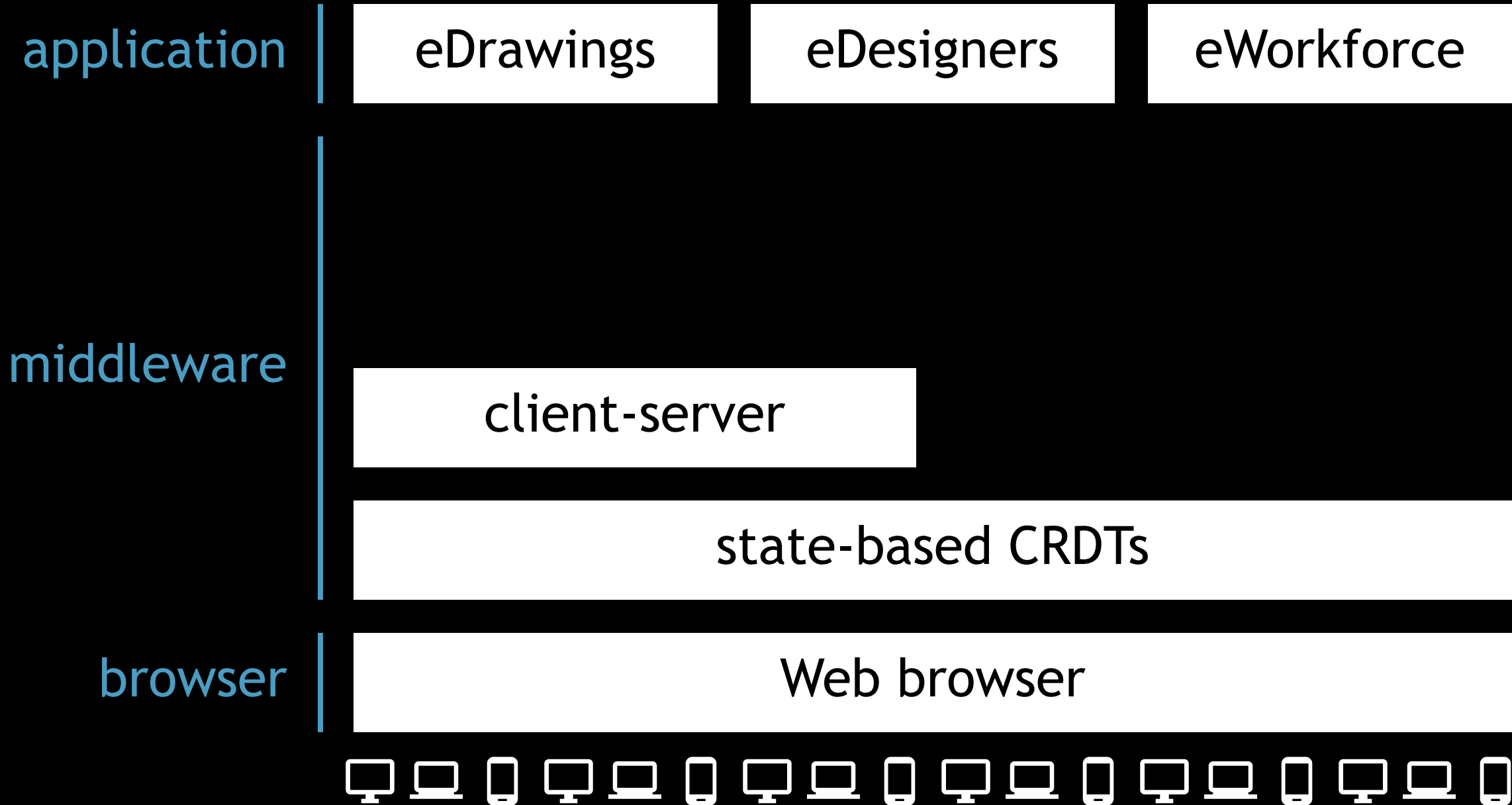
state-based CRDTs

browser

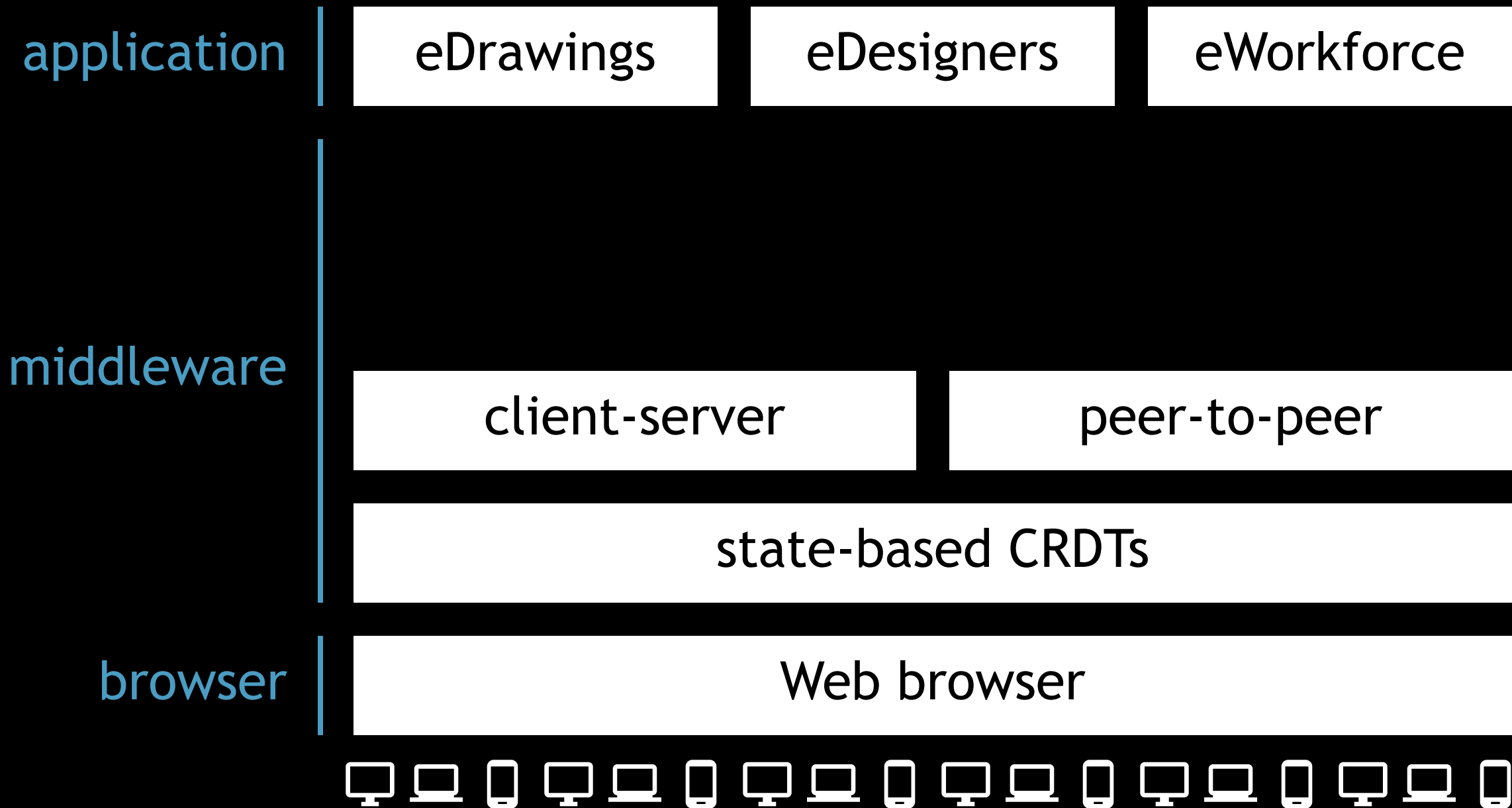
Web browser



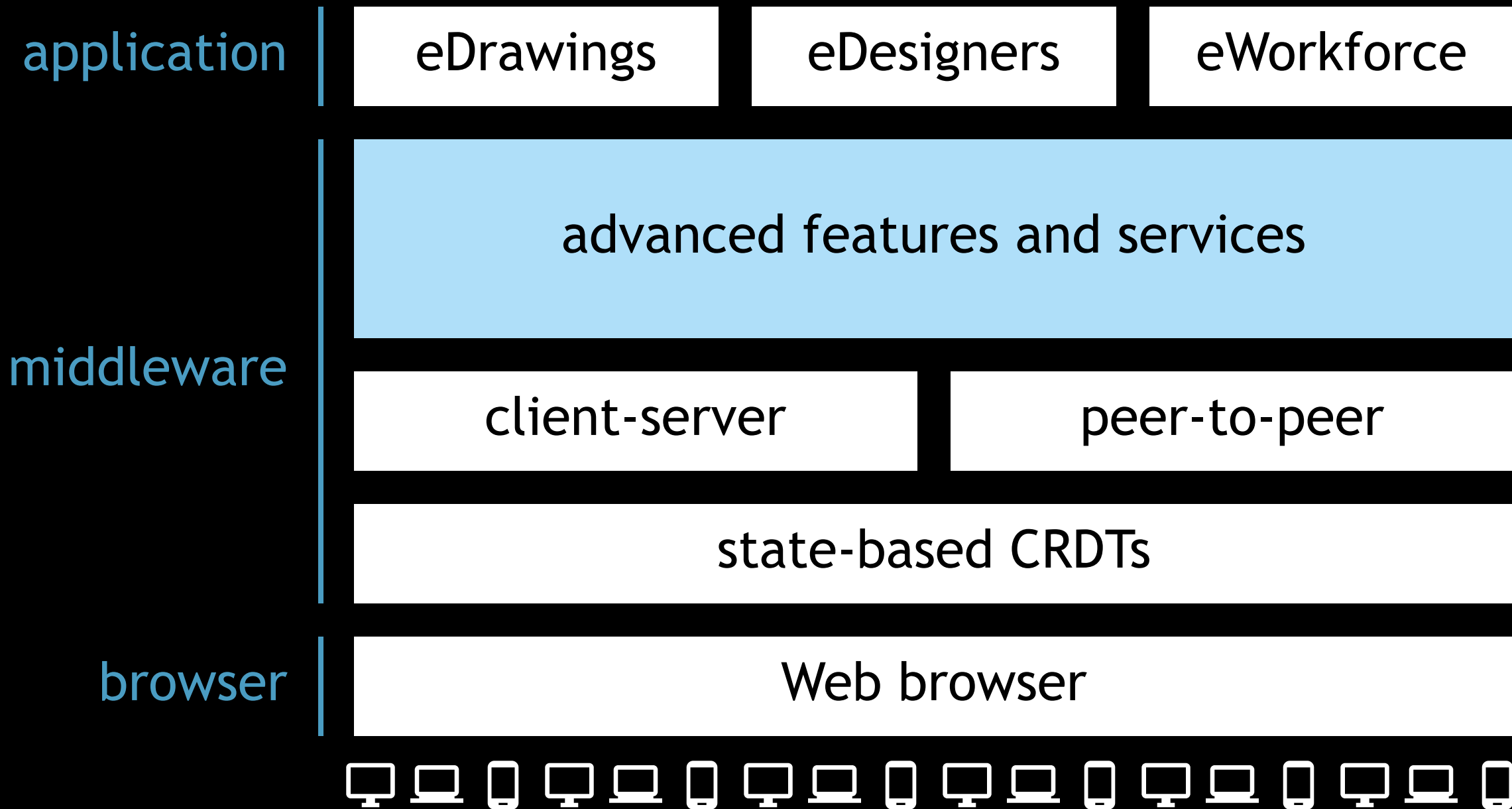
Client-centric web middleware using CRDTs



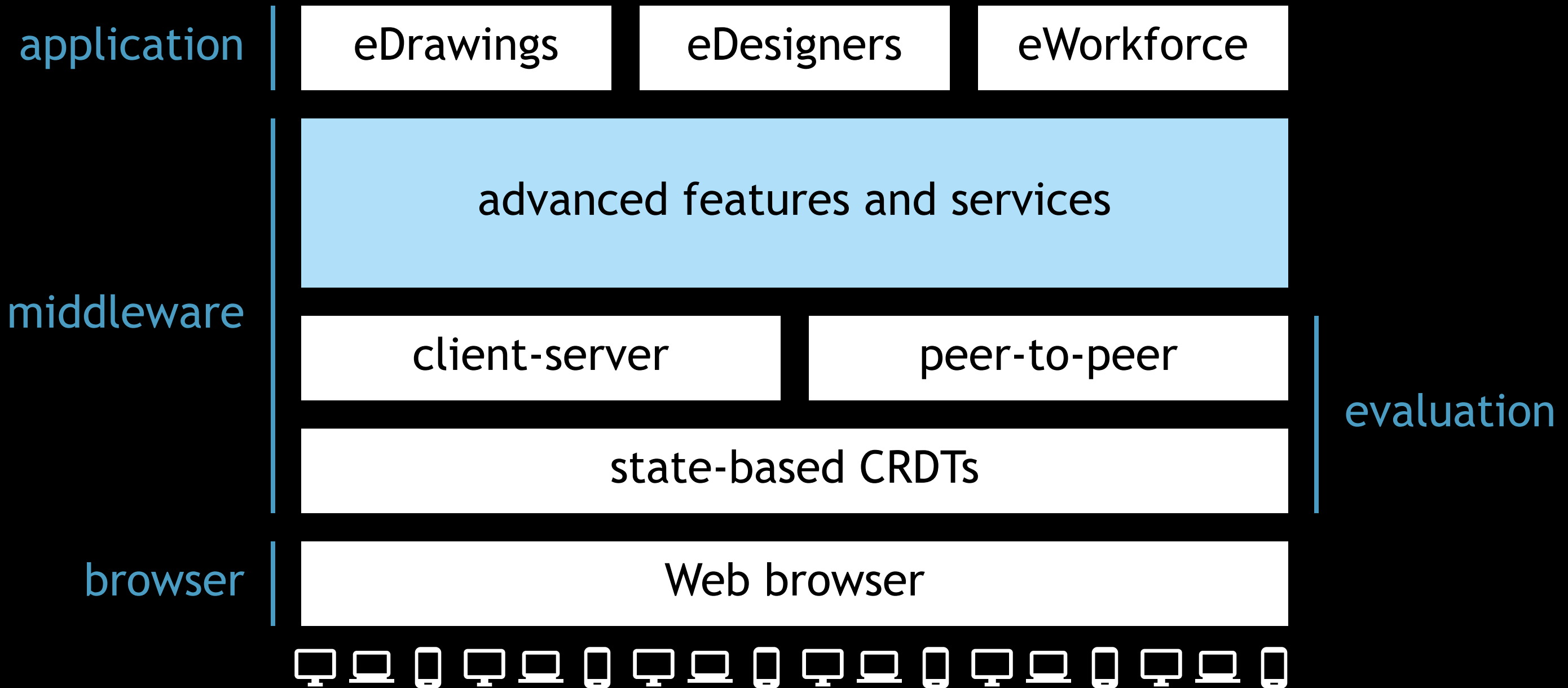
Client-centric web middleware using CRDTs



Client-centric web middleware using CRDTs



Client-centric web middleware using CRDTs



30 distributed clients

100 ms latency

1 update / second / client

Peer-to-peer increases interactivity

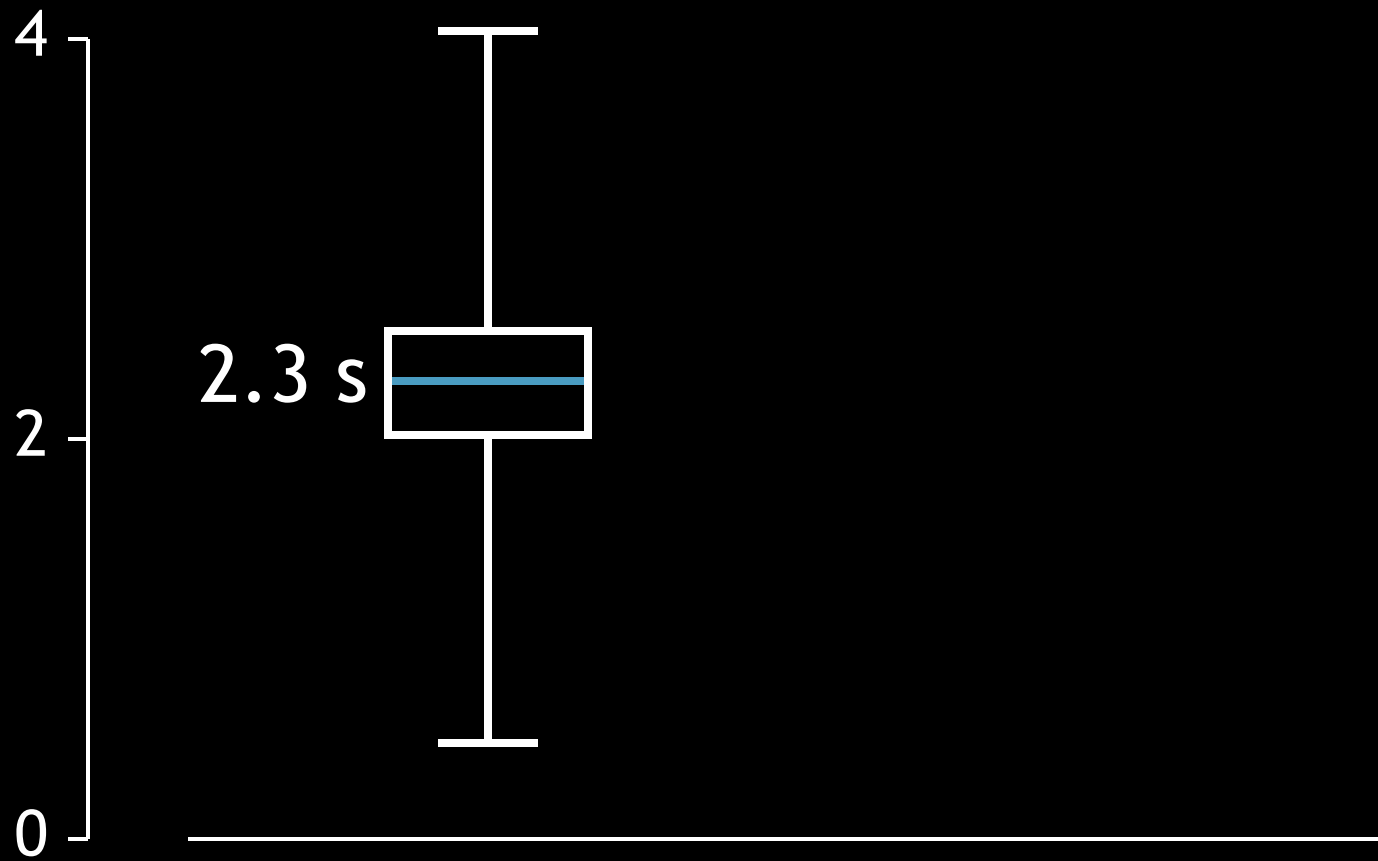
30 distributed clients

100 ms latency

1 update / second / client

Peer-to-peer increases interactivity

client-server

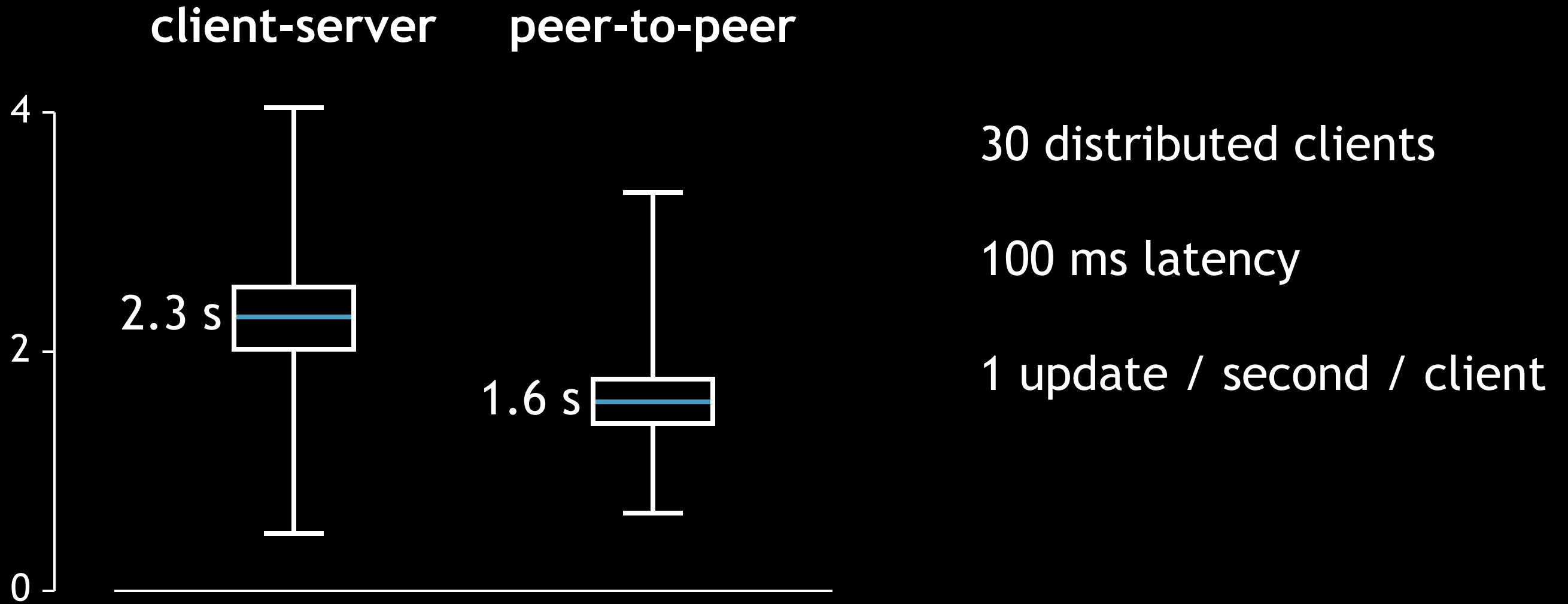


30 distributed clients

100 ms latency

1 update / second / client

Peer-to-peer increases interactivity



Required bandwidth is available

client-server peer-to-peer

0.3 Mbit/s

1.2 Mbit/s

Required bandwidth is available

client-server

peer-to-peer

US mobile

0.3 Mbit/s

1.2 Mbit/s

27 Mbit/s



Related work

Related work: advanced requirements are missing

GUI-focused data binding

React, Angular

Client-server REST

jQuery

Data-synchronization

PouchDB, Yjs, Legion

Summary




Motivational case studies

Identified 7 requirements




Browsers are ready (more or less)

Related work

Future work before the browser will become the application server of the future

-  User-centric access control
-  Distributed queries
-  Decentralized processing and analysis

One day, We will have decentralized web applications

-  Which will work on any device
-  Whether you are connected to internet or not
-  Basis for putting users into control of their data

The Web Browser as Distributed Application Server: Towards Decentralized Web Applications in the Edge

Kristof Jannes, Bert Lagaisse, Wouter Joosen

DistrINet

KU LEUVEN