

FOSDEM 2021

Making Nextcloud Hub COOL(er)

Implementing out of the box collaboration

By Michael Meeks

General Manager, Collabora Productivity

@mmeeks @CollaboraOffice



“Stand at the crossroads and look; ask for the ancient paths, ask where the good way is, and walk in it, and you will find rest for your souls...” - Jeremiah 6:16

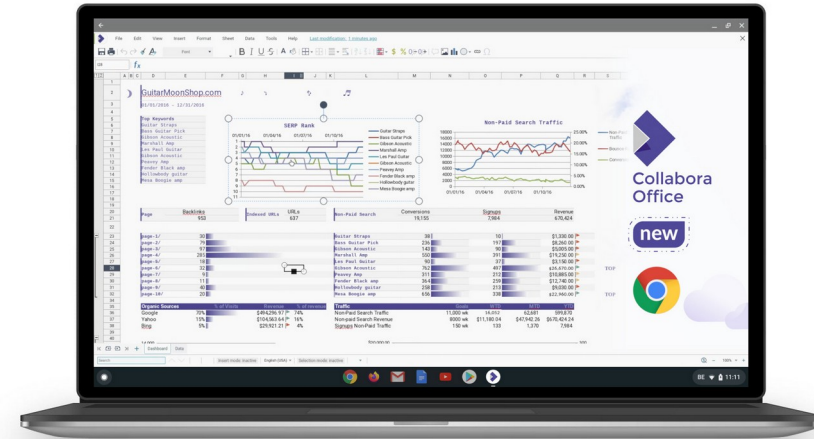
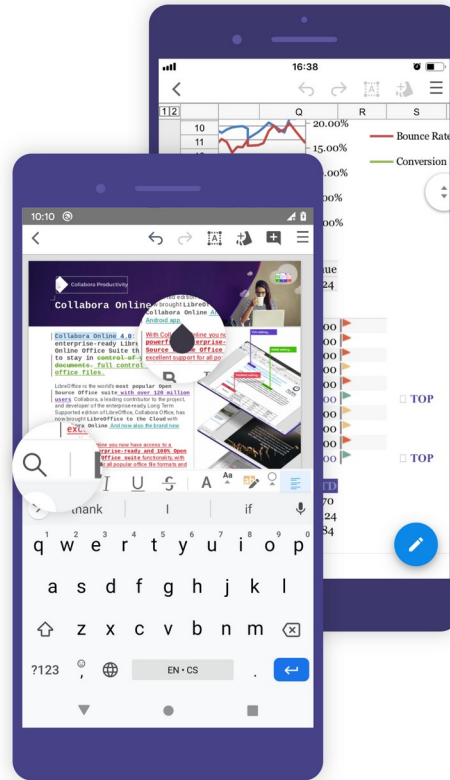
Background.



Collabora Online

Built with awesome LibreOffice Technology

Rich, interoperable,
collaborative editing,
everywhere.



An amazing PHP application

Nextcloud Hub



Collabora
Online

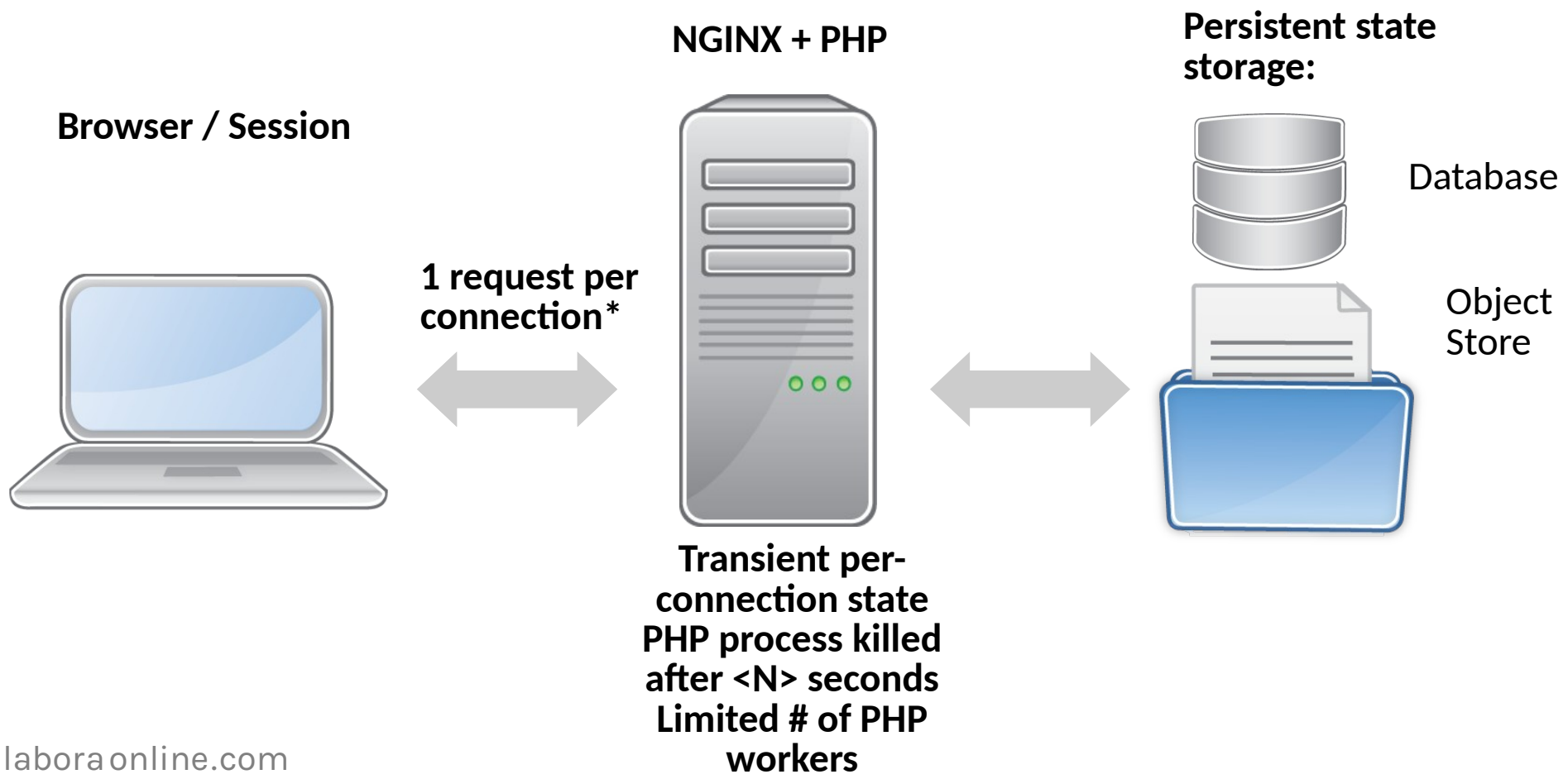


A bundle of collaborative applications you can choose as you install the product.

**Problem: initially no
Collabora Online goodness
in Nextcloud Hub !**

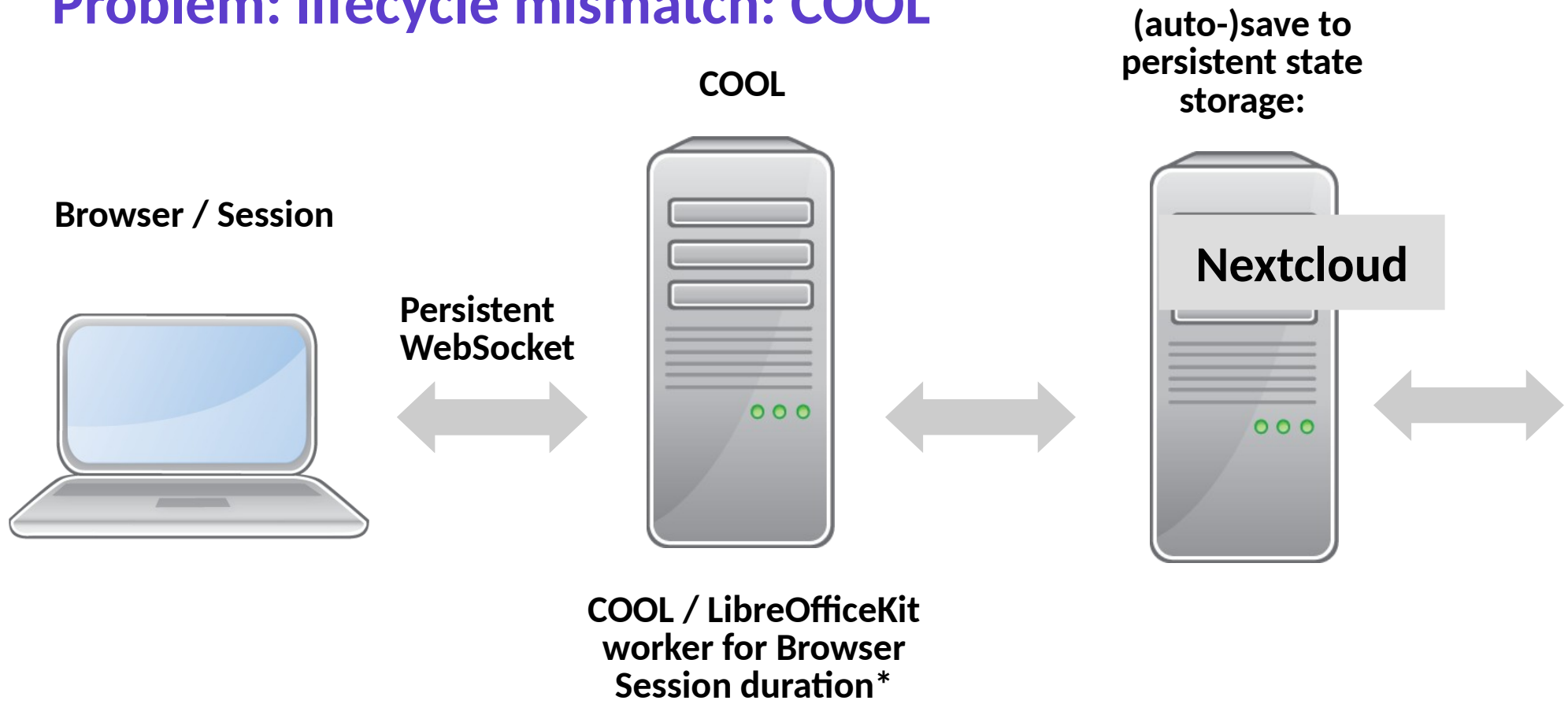


Problem: lifecycle mismatch: Nextcloud





Problem: lifecycle mismatch: COOL





The problem:

PHP

- Obviously no WebSocket support by design
- Nginx (& Apache): Half-duplex, one-shot
 - One data slurp browser → web-server → PHP
 - One data slurp PHP → web-server → browser
- Re-generates all state to answer request from scratch on each request

Collabora Online

- Requires a persistent connection to the browser
- Can't re-load the doc: ~250ms + document fetch and save ~100ms + document store for each key-stroke / event.

**Some Attempts to
get the goodness to
more people**
over the last year



Quick try-out: built-in demo servers ...

Please make sure you understand that the following will happen if you set up the Collabora Online demo.

- The service will send users documents to Collabora and/or third party demo servers.
- This service is not intended for production use, hence the documents will show tile watermarks.
- The demo service may be under heavy load, and its performance is not representative in any way of the performance of an on-premise installation.
- These servers are used for testing and development, and may run test versions of the software. As such they may crash, burn, and re-start without warning.
- The users documents will not be retained by a third party after their session completes except in exceptional circumstances. By using the service, the user gives permission for Collabora engineers to exceptionally use such document data, solely for the purpose of providing, optimizing and improving Collabora Online. Such document data will remain confidential to Collabora and/or any third party providing a demo server.

At the first use and after an update, each user will get the warning, explaining all the above.

I agree, and use the demo server

I will setup my own server

Use a demo server

You can use a demo server provided by Collabora and other service providers.

Select a demo server

- Friprogramvarusyndikatet – Sweden**
- Collabora Productivity Ltd. – Ireland
- Collabora Productivity Ltd. – North California, USA
- Collabora Productivity Ltd. – Tokyo, Japan
- Collabora Productivity Ltd. – São Paulo, Brazil



Built-in demo servers: iff safe & route-able ...

● Use a demo server

You can use a demo server provided by Collabora and other service providers for giving Collabora Online a try.

Your NextCloud setup is not capable of connecting to the demo servers because:

- *it is a local setup (localhost)*
- *it uses an insecure protocol (http)*

For use cases like this, we offer instructions for a [Quick tryout with Nextcloud docker](#).

Can we do better ?



Getting a persistent background application ...

After experimenting it turns out that PHP allows:

```
exec("bash -c \"( $applImage || $applImage --appimage-extract-and-run )  
>/dev/null & disown\"")
```

This allows us to launch a persistent app-image of COOL

- Whoot: only requirements are:
 - A pre-configured AppImage of COOL → for your architecture
 - The ability to download that (largeish) in a given time ...

Need to be careful, manage pids, re-start etc.

cf. <https://github.com/CollaboraOnline/richdocumentscode>

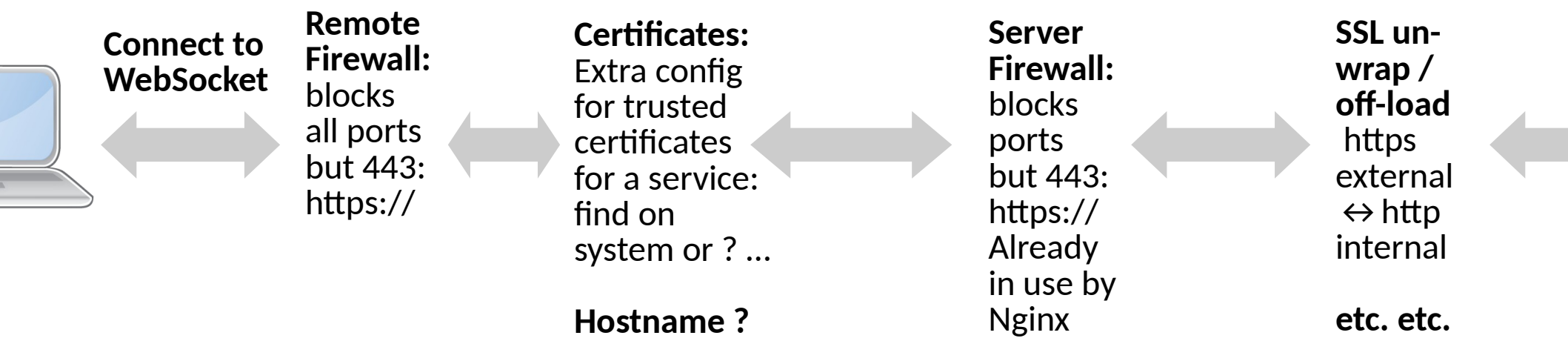


Problem: Talking to that background app

In theory - could open & expose a public websocket

- Browser could quickly connect directly to it !

Serious unpredictable topology problems make this ~impossible





Solution:

Re-use existing PHP configuration that 'just works'

- drop a proxy.php into the live/working Nextcloud
- connect to local port 9982 to talk to coolwsd

Avoids ~all complexities of topology !

- “Just pass the data to/fro ...”

Problems:

- PHP proxy prototype ~3ms per request proxied.
- Add in Nextcloud PHP / plugin infrastructure: ~110ms per request
 - Possible to configure caching / optimizing servers etc. but ... [!]
- Nextcloud kindly added a cool shaped plugin for their security model.



PHP Problems: “just pass the data to/fro”

PHP: Designed to make things easy for normal cases.

- We want the ‘raw’ un-processed data streams
 - **php://input** → sounds promising – but is not.
 - No headers, unwanted escaping*
 - rfc1867 → content (insert image eg.)*
 - split into separate \$_FILES*
 - **So:** re-build the headers & rfc1867 content that back into a raw data-stream in PHP [!]
- **php://output** → can’t output headers only content.

Minor Advantage: can inject our own ProxyPrefix header easily enough ...



PHP Problems #2: “just pass the data to/fro”

PHP: half-duplex, can't stream in and out easily.

- So no real async I/O support
 - why bother with that ?
- Blocking reads, then blocking writes

PHP: sockets – love to use Unix Domain Sockets

- Faster
- no public TCP socket & need to check origin is local
- Requires non-widely installed PHP module people don't have.

An ApplImage ...



Appimage issues ...

Security: home use-ifying ...

- Running un-privileged, without chroot isolation
- No known exploits, but ...

Bundling

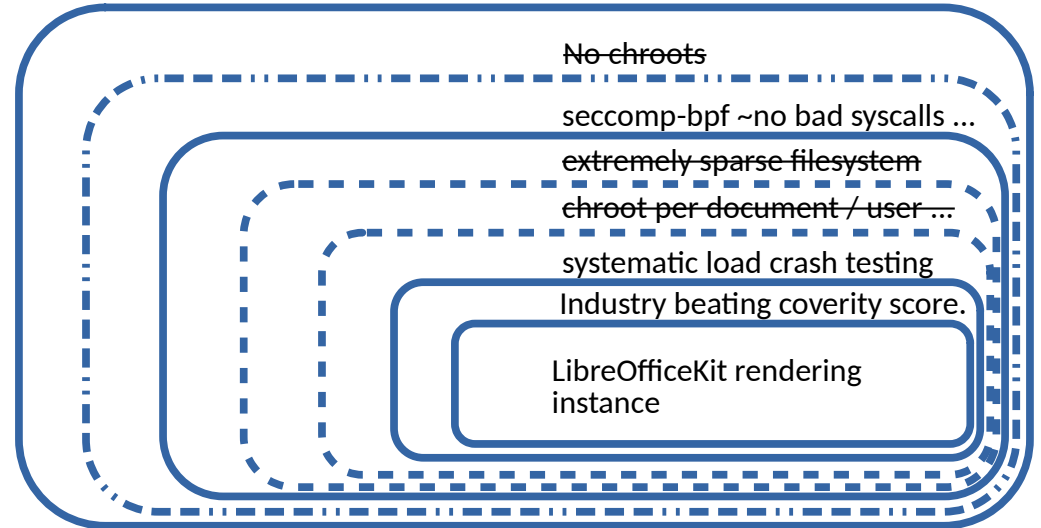
- System vs. bundled libraries & data
- fonts / dictionaries etc.

Architecture checks → 64bit platforms only ...

Lifecycle

- Version information for upgrade & clean restart

Shared Nextcloud's VM / Docker Container





coolwsd side



Online socket code lifecycle re-work:

WebSocketHandler

- base-class for sessions / in Kit & WSD processes

MessageHandlerInterface

- New base-class for Kit, DocumentBroker etc.
- handleMessage / hasQueuedMessage / writeQueuedMessages

ProtocolHandlerInterface ↔ Socket abstraction

- setMessageHandler / sendText/BinaryMessage / shutdown
- Parent of: WebSocketHandler, ProxyProtocolHandler

Major lifecycle re-work



WSD socket re-work...

wsd/ProxyProtocol.cpp

* The ProxyProtocol creates a web-socket like connection over HTTP

* requests. URLs are formed like this:

* 0 1 2 3 4 5

* /lool/<encoded-document-url>/ws/<session-id>/<command>/<serial>

* <session-id> can be 'unknown'

* <command> can be 'open', 'write', 'wait', or 'close'

Javascript ...



Javascript ... frontend

New Proxy socket alternative.

- Parse the new protocol, queue and emit events.
- Throttle input / output
- Several generations of proxy implementation.

Innumerable URI related problems

- Lots of URLs changed, created helpers for JS
- CSS → a major problem, walk all of it & re-write URLs in JS
 - Urk ...

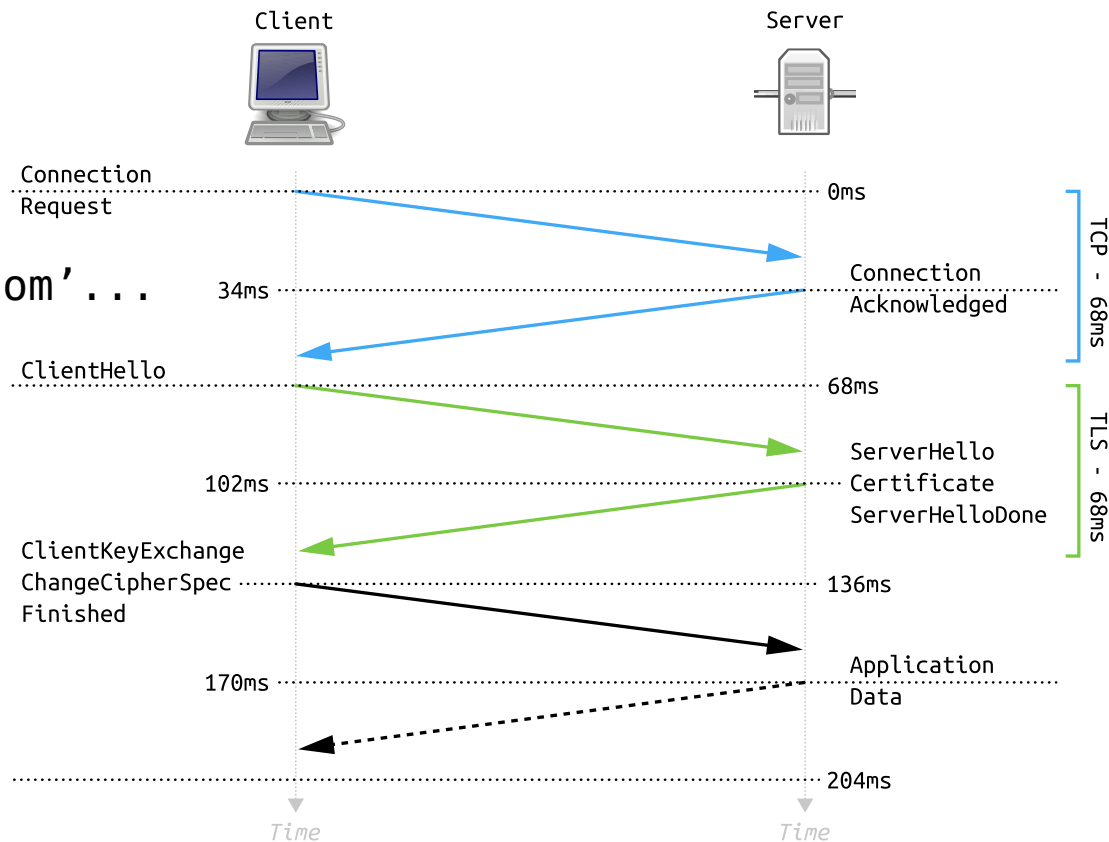


Javascript - XMLHttpRequest API

The API that launched 'Web 2.0'

```
var req = new XMLHttpRequest();  
req.open('POST', 'https://foo.com' ...  
req.send(log);
```

- How can it be fast enough to approach websockets ?
- A single TLS handshake has two round-trips ... before you send any data!



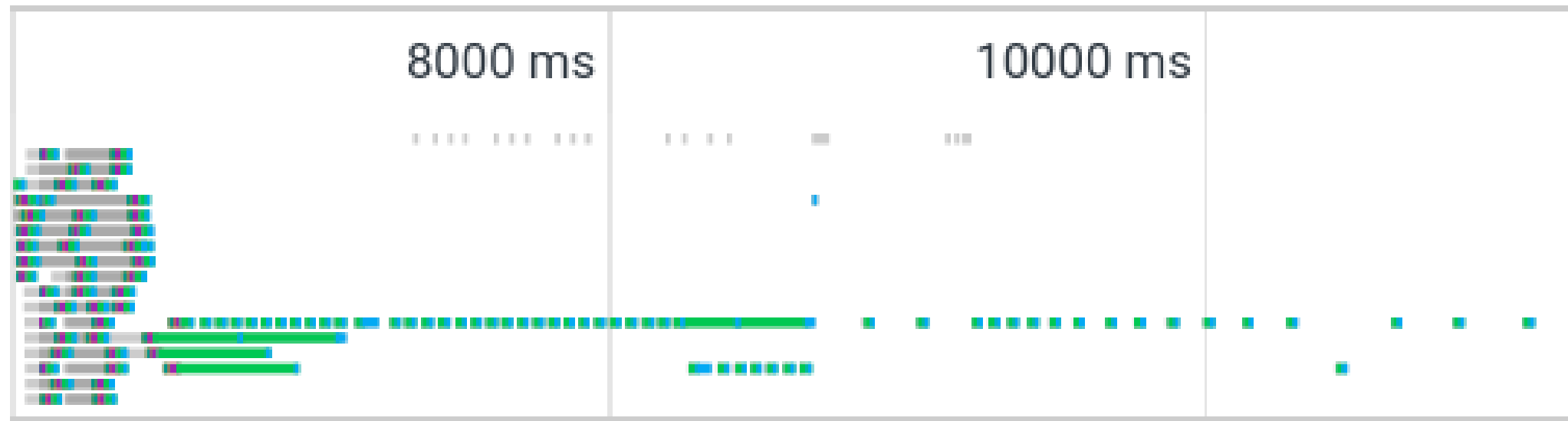
Performance



How can this possibly scale ?

Rescued by **Persistent Connections**

- Connection: keep-alive → The magic header. Not tying up a PHP worker
- see below, 4 cnxs going down to 1 when idler



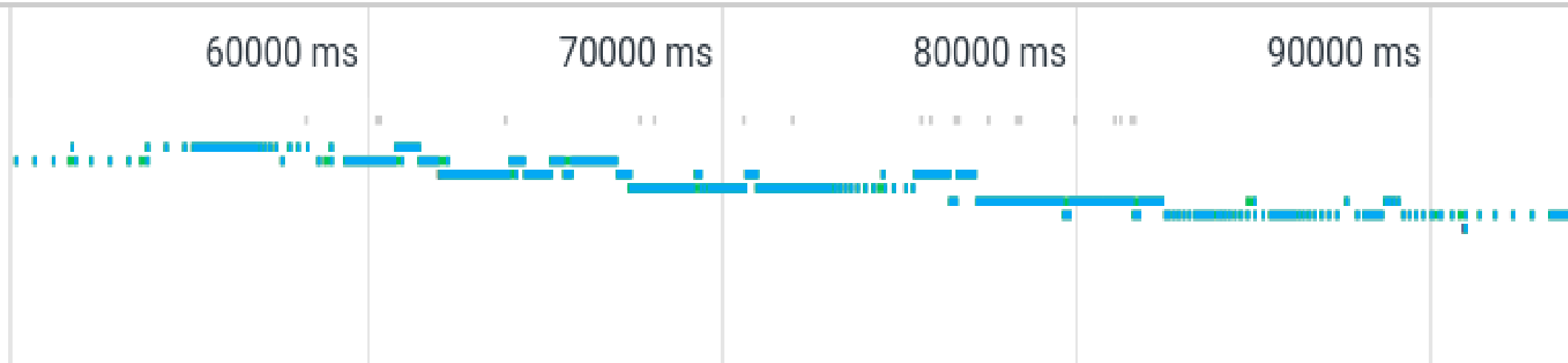


How can this possibly scale ? #2

Exponential back-off

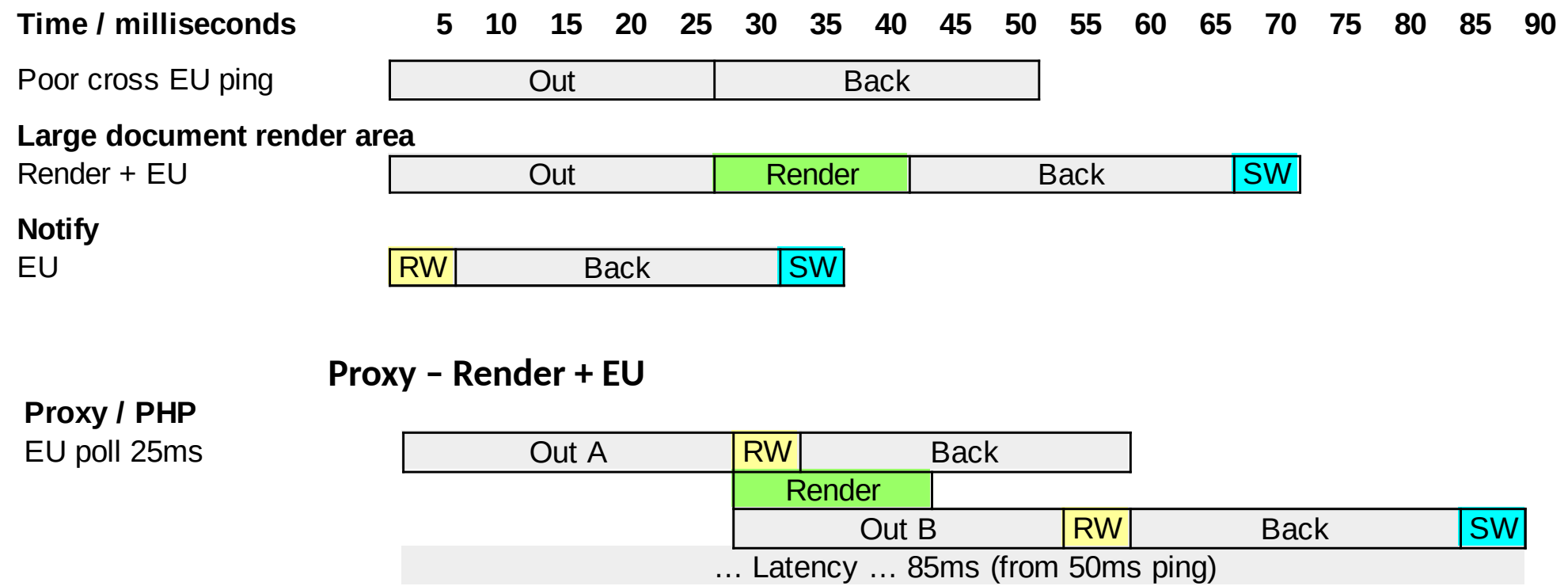
- If we are not typing, and we don't get any interesting events when we poll
 - Poll less frequently – back-off to 500ms waits: 2x per second.

Slow closure & rotation of kept-alive connection every 5-15 seconds





Performance: surprisingly good in-continent.

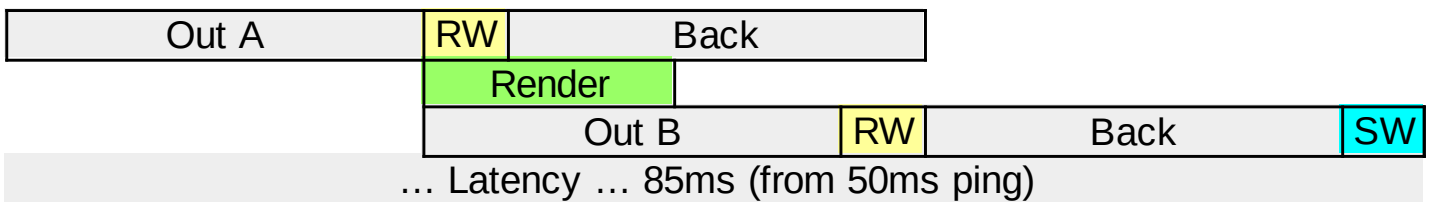




What about long polling ?

Proxy - Render + EU

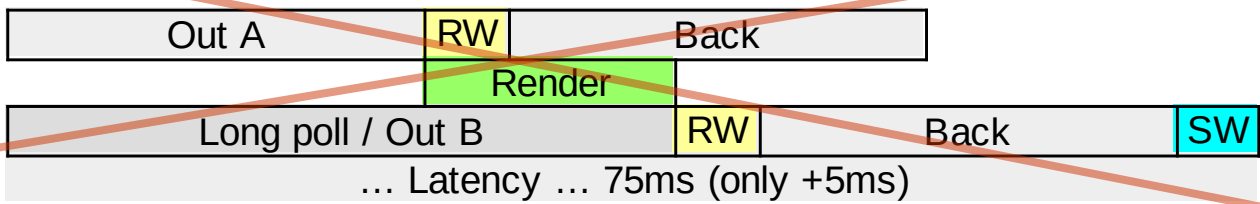
Proxy / PHP
EU poll 25ms



Proxy - Render + EU - hypothetical long-poll (as implemented)

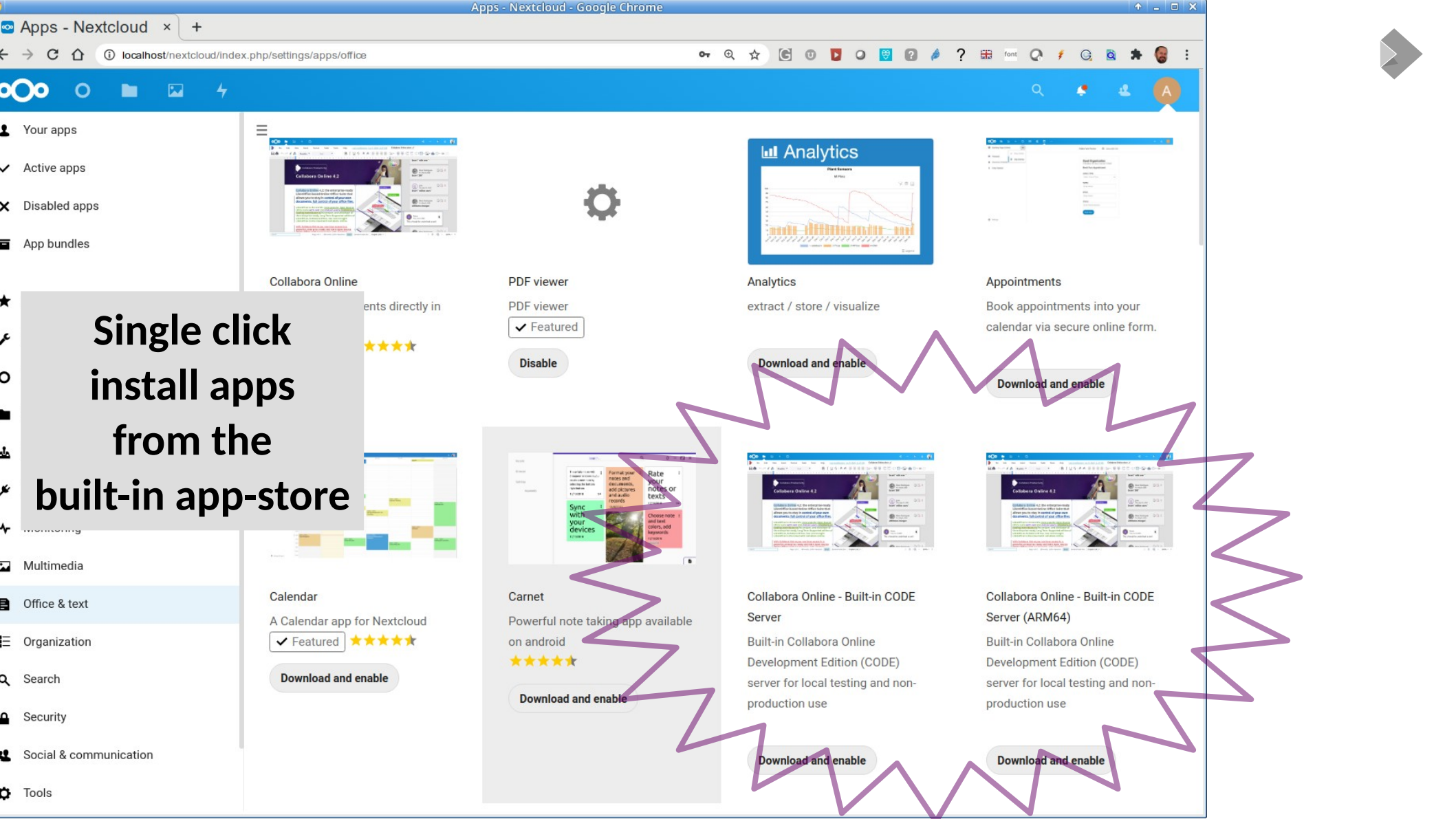
Proxy / PHP
EU poll 25ms

Long polling socket:



Sadly not - your webserver has only ~10 PHP worker threads at any one time
In & out fast - is mandatory ... to scale to many users.

Finally:



- Your apps
- Active apps
- Disabled apps
- App bundles
- Multimedia
- Office & text
- Organization
- Search
- Security
- Social & communication
- Tools

Single click
install apps
from the
built-in app-store



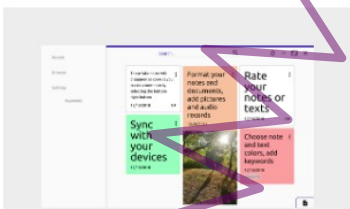
Collabora Online
Collabora Online 4.2
★★★★★



Calendar
A Calendar app for Nextcloud
★★★★★
Download and enable



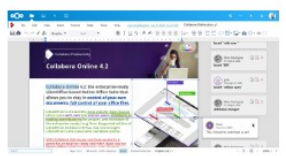
PDF viewer
PDF viewer
✓ Featured
Disable



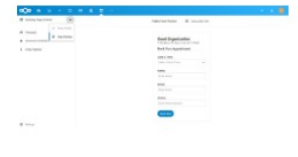
Carnet
Powerful note taking app available on android
★★★★★
Download and enable



Analytics
extract / store / visualize
Download and enable



Collabora Online - Built-in CODE Server
Built-in Collabora Online Development Edition (CODE) server for local testing and non-production use
Download and enable



Appointments
Book appointments into your calendar via secure online form.
Download and enable



Collabora Online - Built-in CODE Server (ARM64)
Built-in Collabora Online Development Edition (CODE) server for local testing and non-production use
Download and enable



A simple one-click install:

Use your own server

Collabora Online requires a separate server acting as a WOPI-like Client to provide editing capabilities.

Use the built-in CODE - Collabora Online Development Edition

Easy to install, for personal use or for small teams. A bit slower than a standalone server and with limited features.

Please upgrade to a faster, better, native server soon.

Make Open Source ROCK

Making things easy for
the user: many, many
months of hard work &
investment.

Not easy.

Many thanks to:
Julius Härtl
Jan Holesovsky
Mert Tumer
Szymon Kłos
Andras Timar
Tamas Zolnai
Ashod Nakashian



Thank you! & Questions

By Michael Meeks

@mmeeks @CollaboraOffice

CollaboraOffice.com

CollaboraOffice.com/CODE

michael.meeks@collabora.com

**Open Source,
Open First**

Oh, that my words were recorded, that they were written on a scroll, that they were inscribed with an iron tool on lead, or engraved in rock for ever! I know that my Redeemer lives, and that in the end he will stand upon the earth. And though this body has been destroyed yet in my flesh I will see God, I myself will see him, with my own eyes - I and not another. How my heart yearns within me. - Job 19: 23-27