



ICALEPCS 2021

18th International Conference on Accelerator
and Large Experimental Physics Control Systems

CANONE3
A NEW SERVICE AND DEVELOPMENT
FRAMEWORK FOR THE WEB AND PLATFORM
INDEPENDENT APPLICATIONS*

Giacomo Strangolino
Lucio Zambon

* inspired by an idea of Alessio I. Bogani



Elettra Sincrotrone Trieste

PUMA

Reliable, secure, scalable and user-oriented design of a multi platform framework based on the most advanced stage of web technologies



HISTORY

from canone to canone3 (PUMA)

- 2006 - **Canone** - Python server, drag and drop designer
- 2016 - **ElettrApp** - Responsive, Cordova, JQuery, Bootstrap
- 2017 - **PWMA** - C++ server + Websocket (named Canone 2), React, React Native
- 2020 – **PUMA** - NChan server + SSE (named Canone 3), interactive SVG, designer tool for responsive pages



SECTION I

THE SERVICE



1. RELIABLE

The system must work

- From any place, time, platform
- Always, regardless the number of clients
- Always, included when part of the system is unavailable
- Included when network performance is slow (or subject to charges!)



WebSocket issues
with proxies...



Events, stream
over channels

Performance independent of the number of clients



2. SECURE

- 1) From external attacks (DoS, intrusion)
- 2) From ill designed clients flooding the service
- 3) Protect the downstream Control System Engine (Tango, EPICS)



- Network architecture
- Security-oriented OSes (BSD!?)
- Framework design, database and code


- No service performance decay
 - Hamper unruly clients



2. SECURE (II)

GOAL secure enough as to avoid VPN or other additional client-side configuration hindering *usability*, especially on mobile devices

EXAMPLE: DDoS



Websocket:
More difficult! :-/
(not HTTP)



Nginx :-)
HTTP

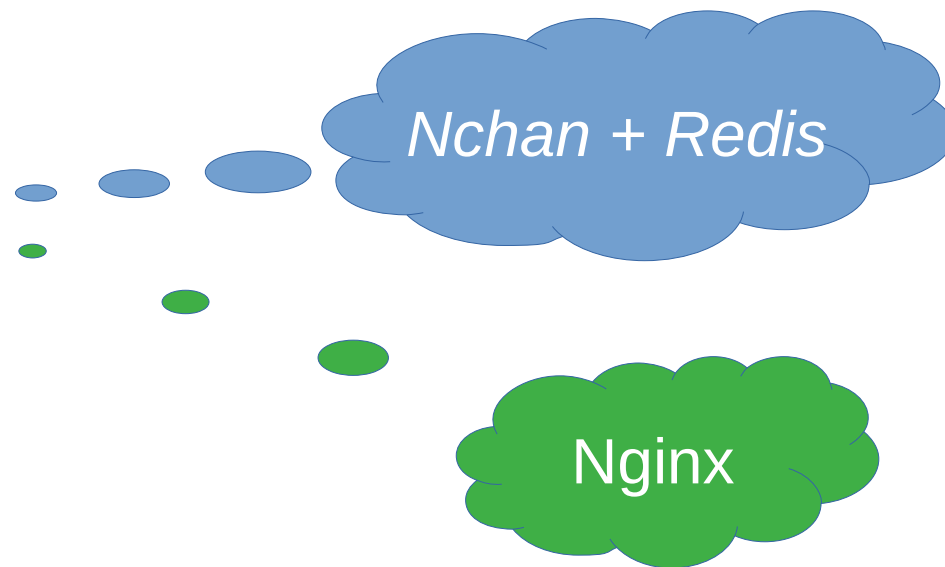


3. SCALABILITY

A good infrastructure must be designed with scalability in mind. It reinforces security and reliability.

1) Horizontal

2) Vertical

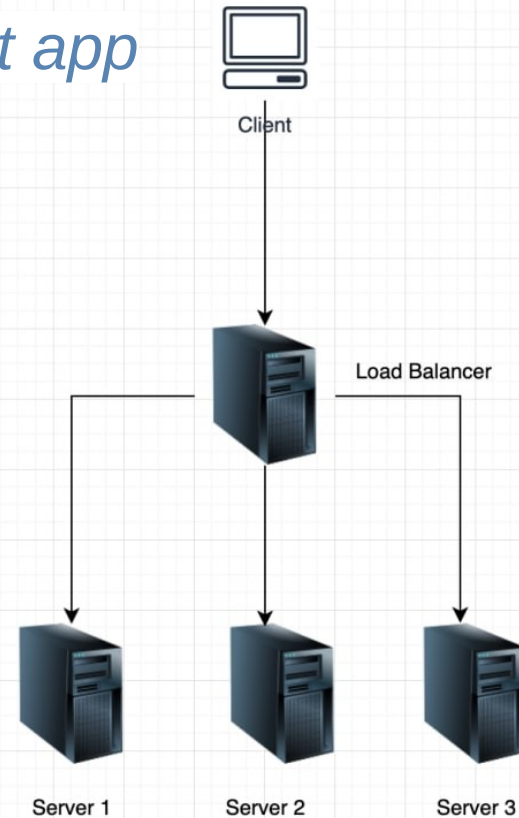


3. SCALABILITY (II)

Platform independent app

nginx

PUMA (canone3) services



Tango, EPICS, ...



4. GENERIC API

Must serve

- 1) The web
- 2) Mobile applications
- 3) Desktop applications *

* Cumbia libs already support the API so that any Qt application can be instructed to rely on either the native control system engine or the HTTP service at runtime





DESIGN RATIONALE

Nginx + nchan + http/SSE * = ?

* inspired by an intuition of Alessio I. Bogani



Nginx + nchan + http/SSE = ?

- ✓ **NGINX**: high performance load balancer, web server and reverse proxy <https://www.nginx.com/>
- ✓ **NCHAN**: flexible *pubsub* for the modern web
- ✓ **SSE**: a server *push* technology: a client receives automatic updates from a server via HTTP connection

=

A scalable, secure, efficient service with multiplexing for web, mobile and desktop applications



Nginx + Nchan + Redis

Redis (Remote Dictionary Server) is an in-memory data structure project implementing a distributed, in-memory key–value database with optional durability.

- ✓ Redis can be used to add data persistence and horizontal scalability, failover and high availability to a Nchan setup.
- ✓ *Redis Cluster* provides a way to run a Redis installation where data is automatically sharded across multiple Redis nodes.

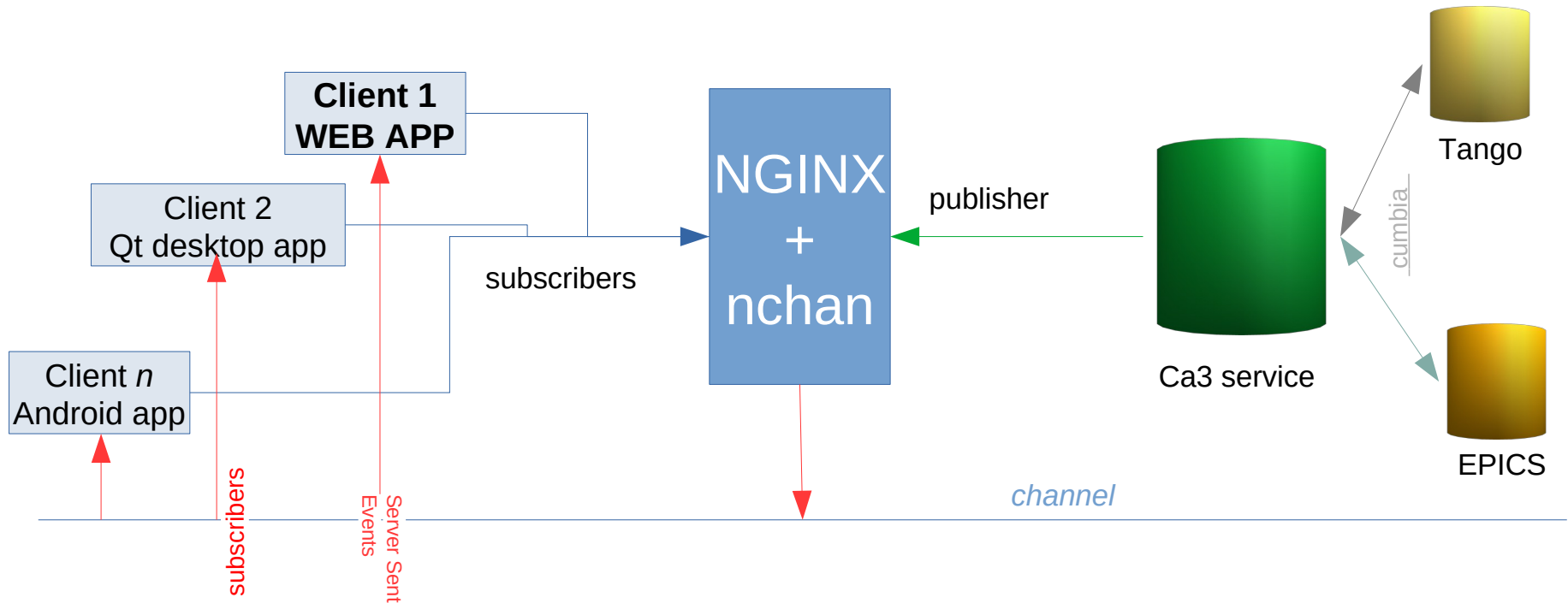
Nchan + Redis

- ✓ add scalability via sharding channels among cluster nodes.
- ✓ *Redis cluster* provides automatic failover, high availability,
- ✓ *Redis cluster* eliminates the single point of failure of one shared Redis server





Nginx + nchan + http/SSE = ?



Additionally

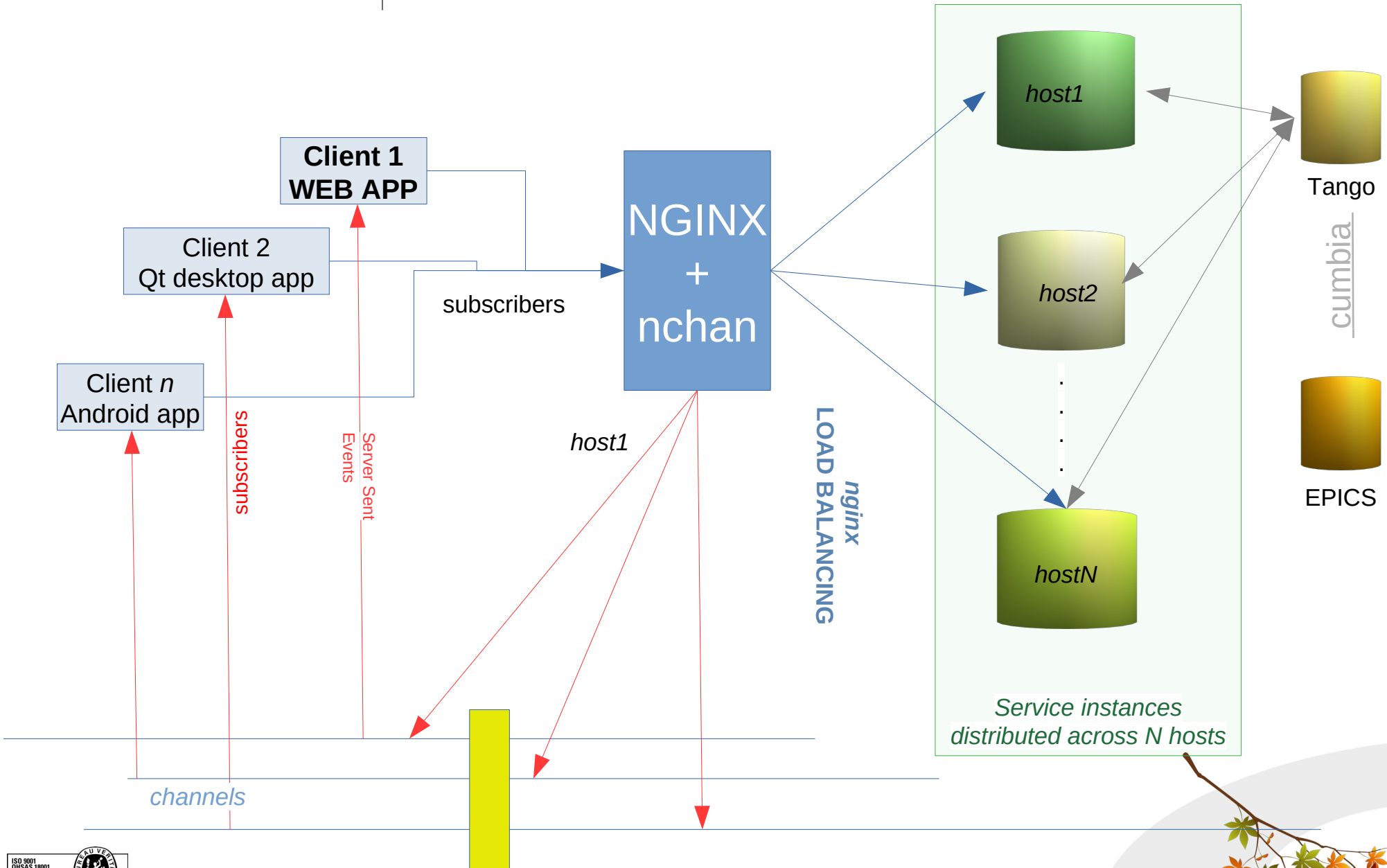
- × Synchronous readings
- × Synch database property fetch
- × Authenticated synchronous writings

Multiplexing: n clients reading x \longrightarrow 1 reader to the native engine



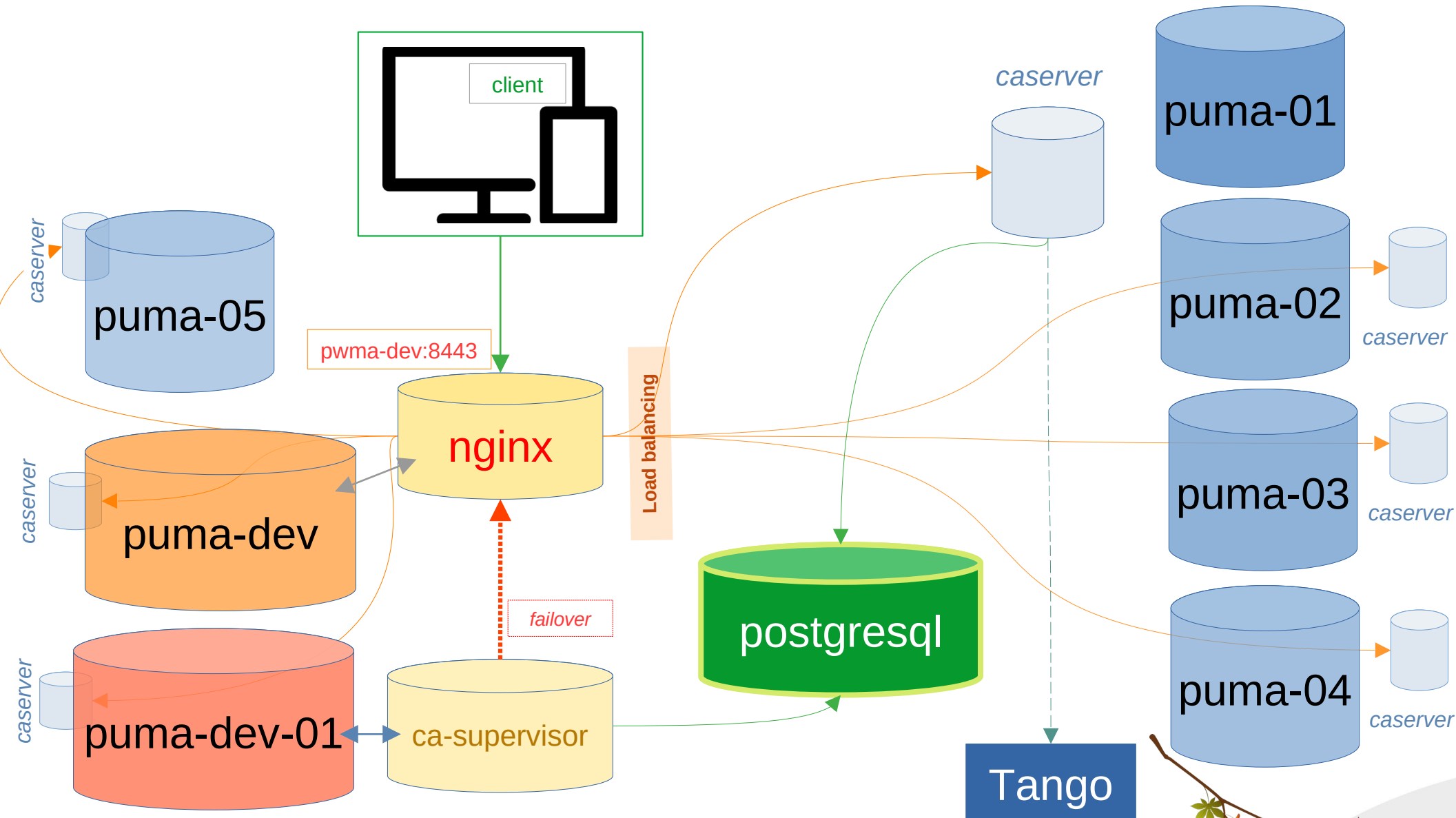


Nginx + nchan – scalability and load balancing







DESIGN RATIONALE - DEPLOYMENT



DESIGN RATIONALE


TEST ENVIRONMENT – LOAD BALANCING

The test environment is pictured in the previous slide. A query to the database shows the the distribution of 1255 readings across six services:



id	started	expected	state	addr	count
311	2021-09-13 13:35:18.331211	2021-09-13 14:20:23.492731	ACTIVE	192.168.205.159	212
320	2021-09-13 13:35:17.06742	2021-09-13 14:20:30.02976	ACTIVE	192.168.205.106	222
326	2021-09-13 13:35:17.653089	2021-09-13 14:20:21.798637	ACTIVE	192.168.205.149	235
327	2021-09-13 13:35:17.54277	2021-09-13 14:20:30.455891	ACTIVE	192.168.205.158	215
324	2021-09-13 13:35:16.584578	2021-09-13 14:20:20.77197	ACTIVE	192.168.205.157	161
328	2021-09-13 13:35:18.383527	2021-09-13 14:20:26.159456	ACTIVE	192.168.205.166	210

(6 rows)



1255



TEST ENVIRONMENT – LOAD BALANCING (II)

Load balancing is accomplished by *nginx*. Take a look at *nginx.conf*:

```
http {
    upstream caserver {
#       hash $http_x_channel consistent;
#
#       A request is sent to the server with the least number of active connections
#       https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/#least\_conn;
#       least_conn;

        server puma-01.elettra.trieste.it:9292 fail_timeout=20s;
        server puma-02.elettra.trieste.it:9292 fail_timeout=20s;
        server puma-03.elettra.trieste.it:9292 fail_timeout=20s;
        server puma-04.elettra.trieste.it:9292 fail_timeout=20s;
        server puma-05.elettra.trieste.it:9292 fail_timeout=20s;
# -dev hosts serve as test nodes
        server pwma-dev.elettra.trieste.it:9292 fail_timeout=20s;
        server puma-dev-01.elettra.trieste.it:9292 fail_timeout=20s;
    }
}
```





DESIGN RATIONALE

TEST ENVIRONMENT – FAILOVER

Simulate a server failure in one of the hosts:

```
puma@puma-04:~$ host puma-04
puma-04.elettra.eu has address 192.168.205.159
puma@puma-04:~$ sudo killall -SEGV caserver
```

Take a look at the *casupervisor* logs:

```
[Mon Sep 13 14:44:17 2021] [thread:0x7f7fc717d700] ca-supervisor: # ACTIVE "puma-dev" 192.168.205.106:9292 | expected: 13s [Mon Sep 13 15:44:30 2021] | started: Sun Sep 13 13:35:17 2021
[Mon Sep 13 14:44:17 2021] [thread:0x7f7fc717d700] ca-supervisor: # ACTIVE "puma-01" 192.168.205.149:9292 | expected: 15s [Mon Sep 13 15:44:32 2021] | started: Sun Sep 13 13:35:17 2021
[Mon Sep 13 14:44:17 2021] [thread:0x7f7fc717d700] ca-supervisor: # ACTIVE "puma-02" 192.168.205.157:9292 | expected: 14s [Mon Sep 13 15:44:31 2021] | started: Sun Sep 13 13:35:16 2021
[Mon Sep 13 14:44:17 2021] [thread:0x7f7fc717d700] ca-supervisor: # ACTIVE "puma-03" 192.168.205.158:9292 | expected: 14s [Mon Sep 13 15:44:31 2021] | started: Sun Sep 13 13:35:17 2021
[Mon Sep 13 14:44:17 2021] [thread:0x7f7fc717d700] ca-supervisor: # ACTIVE "puma-05" 192.168.205.166:9292 | expected: 19s [Mon Sep 13 15:44:36 2021] | started: Sun Sep 13 13:35:18 2021
[Mon Sep 13 14:44:17 2021] [thread:0x7f7fc717d700] ca-supervisor: # ZOMBIE "puma-04" 192.168.205.159:9292 | expected: -103s [Mon Sep 13 15:42:34 2021] | started: Sun Sep 13 13:35:18 2021
```



TEST ENVIRONMENT – FAILOVER (II)

Execute the same query as in the first *load balancing* inspection:

id	started	expected	state	addr	count
320	2021-09-13 13:35:17.06742	2021-09-13 14:47:00.696596	ACTIVE	192.168.205.106	434
326	2021-09-13 13:35:17.653089	2021-09-13 14:46:52.57544	ACTIVE	192.168.205.149	235
327	2021-09-13 13:35:17.54277	2021-09-13 14:47:01.1764	ACTIVE	192.168.205.158	215
324	2021-09-13 13:35:16.584578	2021-09-13 14:47:01.49247	ACTIVE	192.168.205.157	161
328	2021-09-13 13:35:18.383527	2021-09-13 14:46:56.885659	ACTIVE	192.168.205.166	210

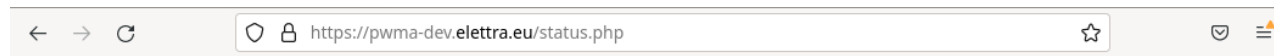
(5 rows)

1255

Please note that load redistribution after a failure is always administered by *nginx*

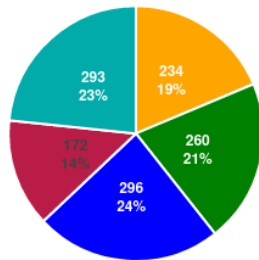


Load balancing and service status: web view



server:port load balancing 2021-09-15 10:45:45 src n: 1255

■ puma-01:9292
 ■ puma-02:9292
 ■ puma-04:9292
 ■ puma-05:9292
 ■ pwma-dev:9292



service history activity history

id	started	expected	state	conf_id	srvnam	addr	port
320	2021-09-15 10:44:12.239299	2021-09-15 10:45:08.870399	ACTIVE	1	pwma-dev	192.168.205.106	9292
317	2021-07-20 10:09:22.534603	2021-07-20 12:27:27.964855	INACTIVE	4	pwma-dev	192.168.205.106	19292
326	2021-09-15 10:44:13.767563	2021-09-15 10:45:05.899086	ACTIVE	5	puma-01	192.168.205.149	9292
324	2021-09-15 10:44:12.994295	2021-09-15 10:45:11.993982	ACTIVE	7	puma-02	192.168.205.157	9292
333	2021-09-14 16:23:37.325481	2021-09-14 16:24:48.04815	ACTIVE	9	puma-03	192.168.205.158	9292
332	2021-09-14 16:23:37.907864	2021-09-15 10:45:11.991821	ACTIVE	10	puma-04	192.168.205.159	9292
328	2021-09-14 16:23:38.346319	2021-09-15 10:45:10.384647	ACTIVE	12	puma-05	192.168.205.166	9292
300	2021-07-06 14:53:22.219985	2021-07-06 15:05:30.619484	INACTIVE	18	woody	192.168.1.159	9292



DESIGN RATIONALE (II)

SECTION II

NATIVE CLIENTS



1. MULTI PLATFORM

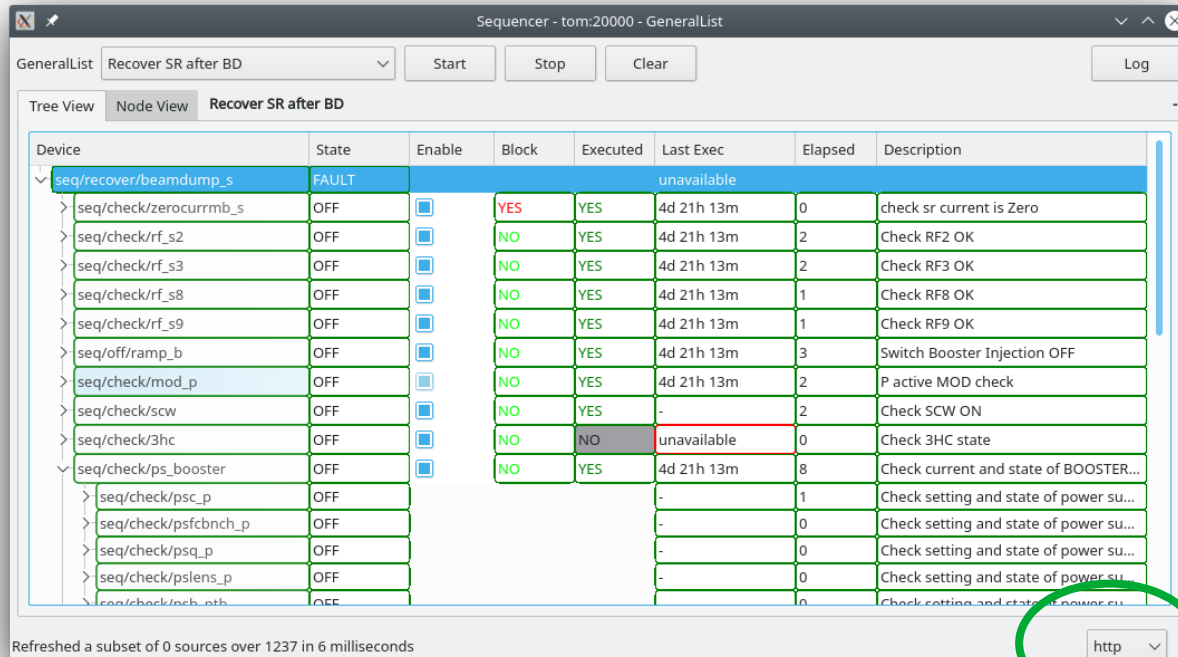
We deem *Telegram* a perfect example of multi platform application:

1. A *native* app on all mobile devices
2. A *desktop* applications for all platforms (even *FreeBSD*)
3. A *web* interface
4. Has a simple and open API to create clients, bots, ...
5. Efficiency, security and privacy centered

These traits have continuously inspired the development of *canone3*

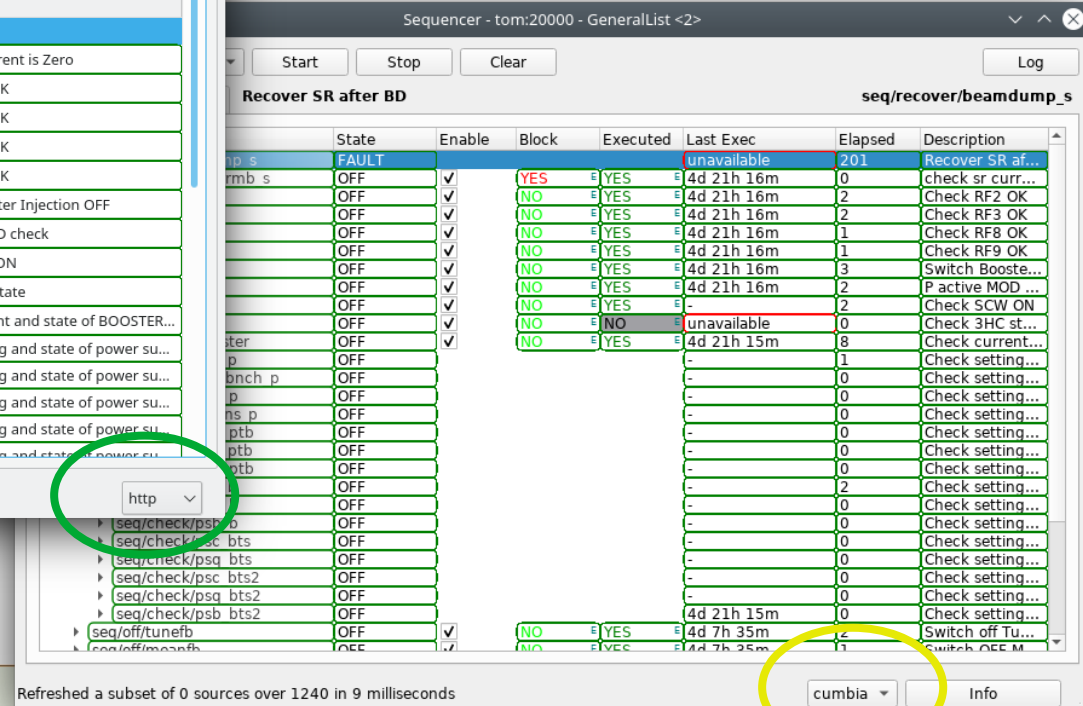


Clients – Qt desktop apps



Device	State	Enable	Block	Executed	Last Exec	Elapsed	Description
seq/recover/beamdump_s	FAULT				unavailable		
seq/check/zerocurrmb_s	OFF	<input checked="" type="checkbox"/>	YES	YES	4d 21h 13m	0	check sr current is Zero
seq/check/rf_s2	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	2	Check RF2 OK
seq/check/rf_s3	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	2	Check RF3 OK
seq/check/rf_s8	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	1	Check RF8 OK
seq/check/rf_s9	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	1	Check RF9 OK
seq/off/ramp_b	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	3	Switch Booster Injection OFF
seq/check/mod_p	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	2	P active MOD check
seq/check/scw	OFF	<input checked="" type="checkbox"/>	NO	YES	-	2	Check SCW ON
seq/check/3hc	OFF	<input checked="" type="checkbox"/>	NO	NO	unavailable	0	Check 3HC state
seq/check/ps_booster	OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 13m	8	Check current and state of BOOSTER...

Refreshed a subset of 0 sources over 1237 in 6 milliseconds



State	Enable	Block	Executed	Last Exec	Elapsed	Description
FAULT				unavailable	201	Recover SR af...
OFF	<input checked="" type="checkbox"/>	YES	YES	4d 21h 16m	0	check sr curr...
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 16m	2	Check RF2 OK
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 16m	2	Check RF3 OK
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 16m	1	Check RF8 OK
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 16m	1	Check RF9 OK
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 16m	3	Switch Booste...
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 16m	2	P active MOD ...
OFF	<input checked="" type="checkbox"/>	NO	YES	-	2	Check SCW ON ...
OFF	<input checked="" type="checkbox"/>	NO	NO	unavailable	0	Check 3HC st...
OFF	<input checked="" type="checkbox"/>	NO	YES	4d 21h 15m	8	Check current...

Refreshed a subset of 0 sources over 1240 in 9 milliseconds

cumbia-http

- ✗ the same app is *designed* and run transparently regardless the engine in use (*native Tango* or *http/SSE*)
- ✗ No *engine-specific* coding
- ✗ Run with the same command line!
- ✗ A virtual machine run at home can reproduce exactly the control room desktop

Clients – Control Room apps

Approaches to running control room applications remotely

Now

- × Control room virtual machines run on the server side
- × Although optimized, a full graphical session is streamed, with a strong impact on *bandwidth* and *battery life* of portable devices. If the former today may not be considered a critical limit (however, estimate the load of a server streaming to hundreds of clients), the latter is indeed a precious resource for portable devices and laptops. *The only relevant piece of information is **data**.*

Then

- × Control room virtual machines run on the client
- × The user runs only the apps he needs
- × The apps run natively (i.e. the fastest possible on the client device), look exactly as in the control room (they are the same) without lags
- × *Only data through the network*, for native and web apps alike



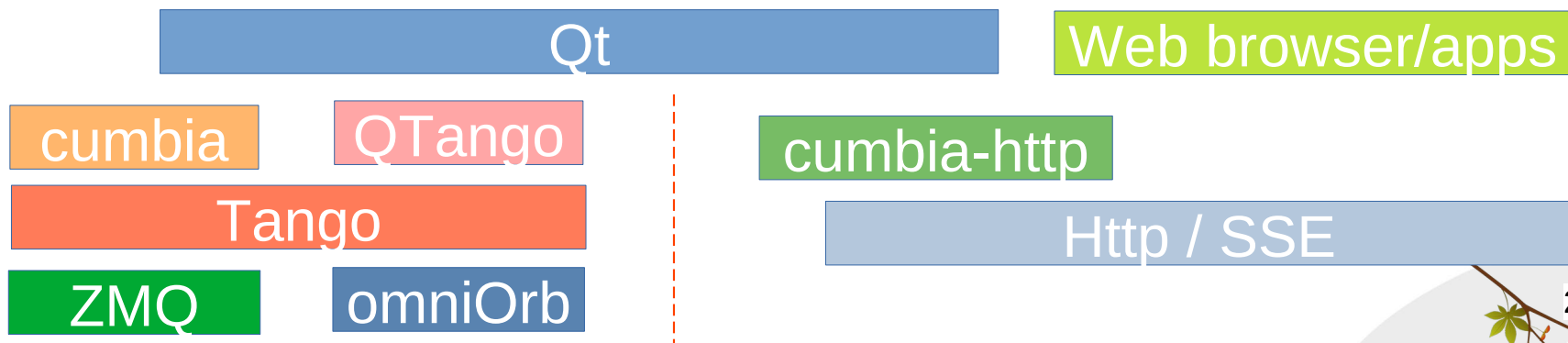
Clients – Control Room apps

Approaches to running control room applications remotely (II)

Additionally

- × Either *virtual* or *physical* machines do not need Tango + dependencies
- × They need only a web browser and Qt for native apps
- × Events only
- × Tango control system potentially *highly relieved* (remember: N clients reading x , 1 read to the control system)

+ all benefits discussed in the *design rationale* section





Elettra
Sincrotrone
Trieste

SECTION III

WEB CLIENTS



Giacomo Strangolino, Lucio Zambon, *ICALEPCS 2021*



1. RESPONSIVE

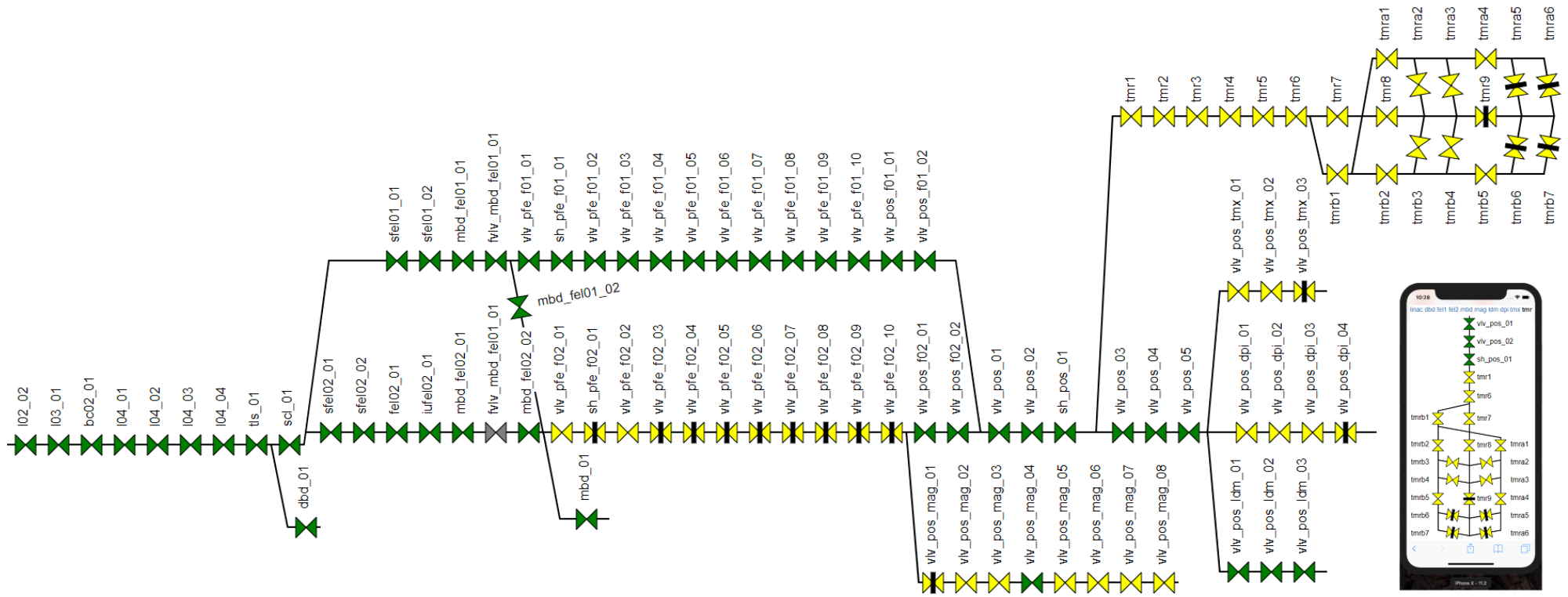
The system must be adaptive

- The application adapts to every device automatically
- On portable devices, screen rotation must not frustrate but rather exploited to increase usability
- Certain actions (especially commands and write operations) are logged





WEB INTERFACES



[youtube.com/watch?v=I-3L0CaJ5X8](https://www.youtube.com/watch?v=I-3L0CaJ5X8)





WEB INTERFACES

GUN - REAL TIME

GUN SAFETY	FIL ABI	FIL CURR	FIL READY	GUN HV
<ul style="list-style-type: none"> SIP40_P1.1 VUOTO INTEGRATO I FILAMENTO LIMITE MASSIMO CHIAVE GUN PORTE GUN FIORETTO GUN BARRA DI TERRA ARIA COMPRESSA FUSIBILI 	<ul style="list-style-type: none"> ABI ACCENSIONE FILAMENTO I FILAMENTO LIMITE MINIMO <p>Curr Fil [mA]</p>	<ul style="list-style-type: none"> FILAMENTO PRONTO <p>Minuti residui</p> <p>1092.25</p>	<ul style="list-style-type: none"> ABI ALIMENT. ALTA TENSIONE CONTROLLO ACCESSI 1 CONTROLLO ACCESSI 2 SIP40_P1.1 VUOTO Istantaneo V GRIGLIA LIMITE MINIMO 80 V CATODO 	<ul style="list-style-type: none"> ABI ALTA TENSIONE GUN ALTA TENSIONE GUN OK ABI GUN VALVOLE SIP40_P1.2 SIP40_P1.3 SIP40_P1.4 SIP300_P1.1 SIP300_P1.2 SIP300_P1.3 SIP300_P1.4

[View Alarms](#)

iPhone X 375 x 812 100% Online

GUN - REAL TIME

logs

- ABI TRIGGER GUN
- GUN HV
- FIL READY
- FIL CURR
- FIL ABI
- ABI ACCENSIONE FILAMENTO
- I FILAMENTO LIMITE MINIMO

Curr Fil [mA]

GUN SAFETY

[View Alarms](#)

youtube.com/watch?v=9TI2S3mThmQ



2. INTERACTIVITY

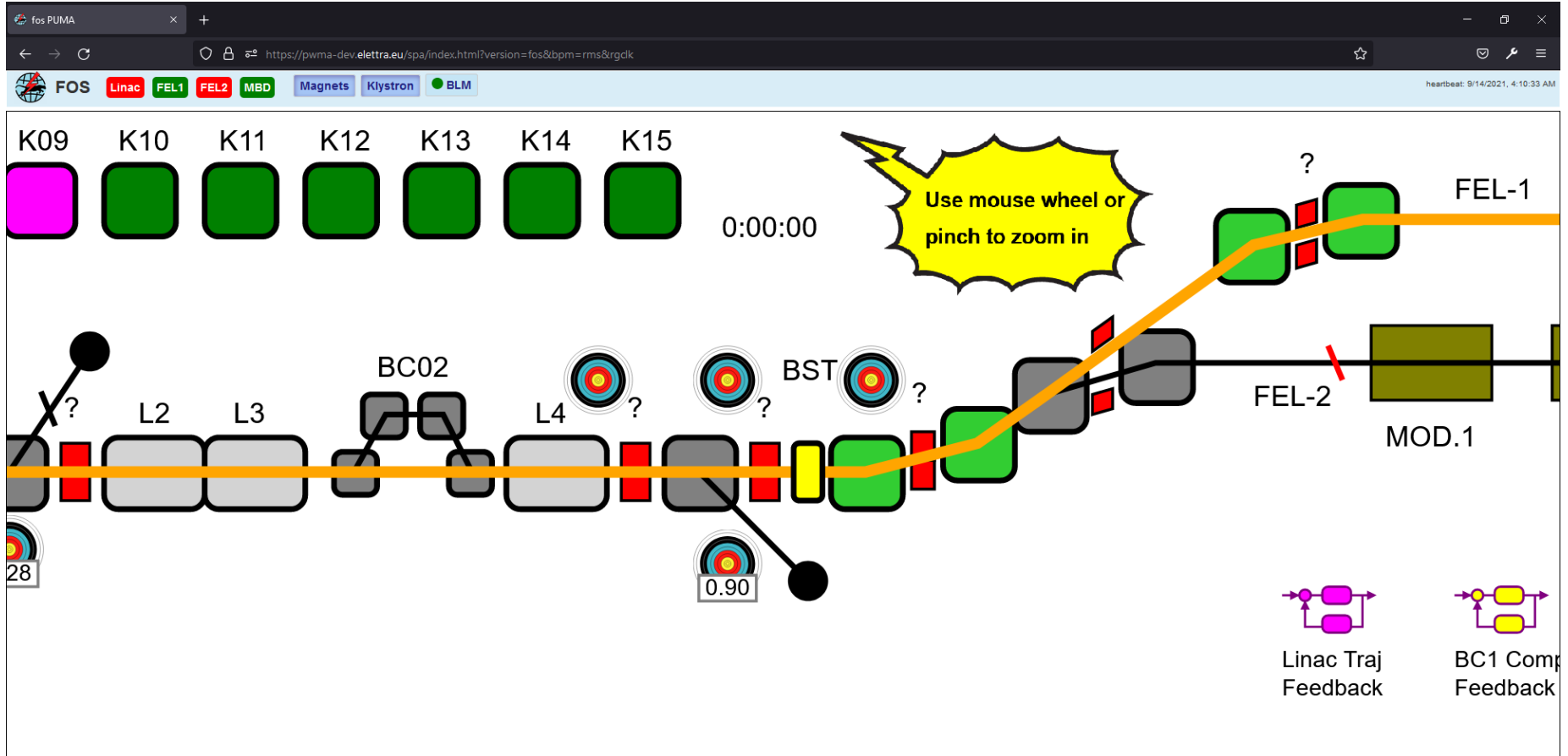
The system must interact with user

- Zoom in and out using mouse wheel on desktop and pinch on mobile
- Depending on zoom level new contents are shown
- Depending on panning and zooming some variable are subscribed/unsubscribed; only the visible area is kept updated





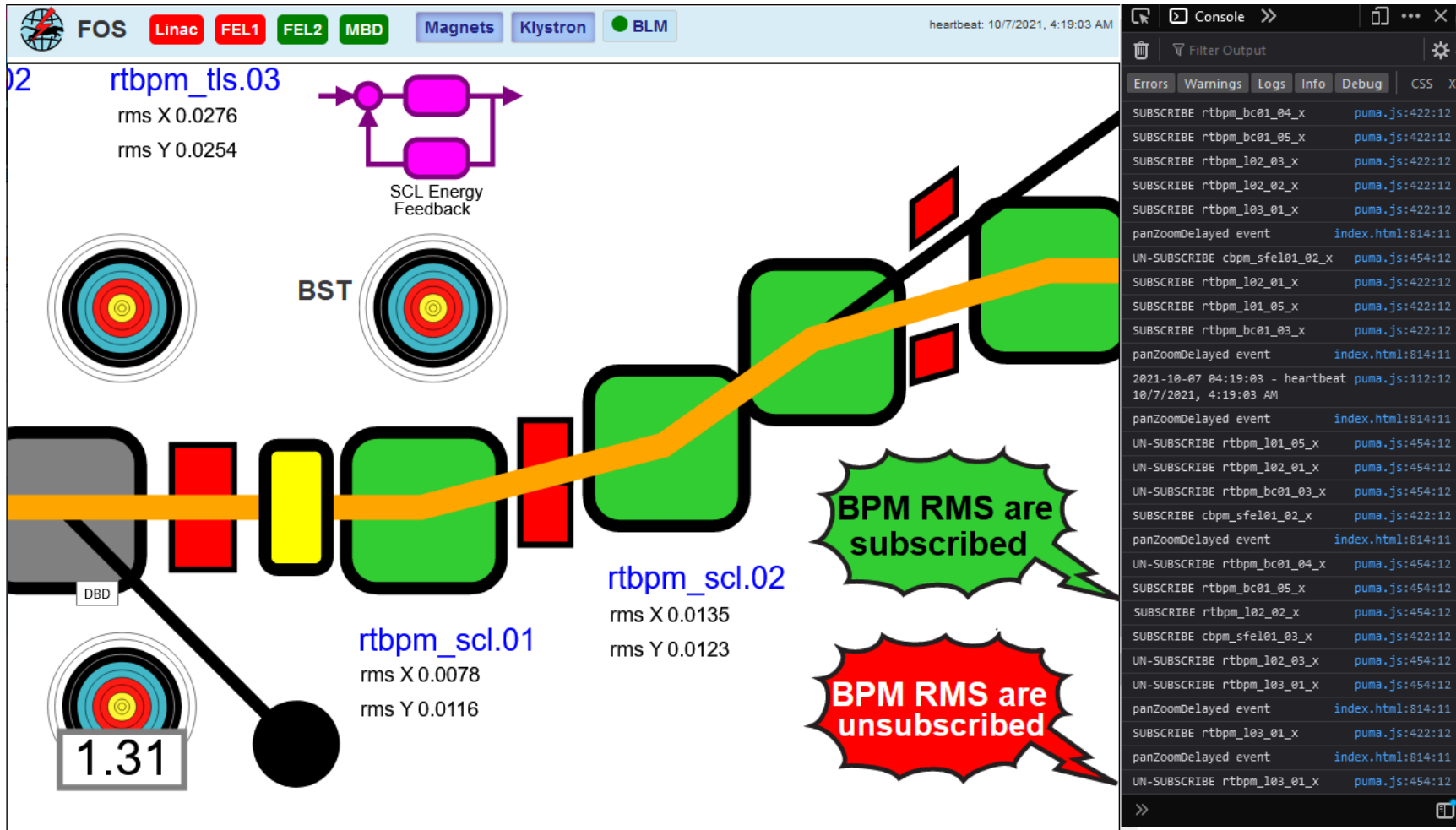
WEB INTERFACES



youtube.com/watch?v=z7FUDB7w2aw



WEB INTERFACES



The screenshot shows a web interface for FOS (Free Oscillation Software) with a navigation bar containing: FOS, Linac, FEL1, FEL2, MBD, Magnets, Klystron, and BLM. The main area displays a beamline diagram with several components and their RMS values:

- rtbpm_tls.03**: rms X 0.0276, rms Y 0.0254
- SCL Energy Feedback**: A feedback loop diagram.
- BST**: Beam Positioning System, represented by a target icon.
- rtbpm_scl.01**: rms X 0.0078, rms Y 0.0116
- rtbpm_scl.02**: rms X 0.0135, rms Y 0.0123
- 1.31**: A component with a target icon and a value of 1.31.
- DBD**: Dipole Bending Dipole, represented by a target icon.

Two callout boxes are present:

- A green callout box: **BPM RMS are subscribed**
- A red callout box: **BPM RMS are unsubscribed**

The console window on the right shows the following log output:

```

SUBSCRIBE rtbpm_bc01_04_x puma.js:422:12
SUBSCRIBE rtbpm_bc01_05_x puma.js:422:12
SUBSCRIBE rtbpm_l02_03_x puma.js:422:12
SUBSCRIBE rtbpm_l02_02_x puma.js:422:12
SUBSCRIBE rtbpm_l03_01_x puma.js:422:12
panZoomDelayed event index.html:814:11
UN-SUBSCRIBE cbpm_sfel01_02_x puma.js:454:12
SUBSCRIBE rtbpm_l02_01_x puma.js:422:12
SUBSCRIBE rtbpm_l01_05_x puma.js:422:12
SUBSCRIBE rtbpm_bc01_03_x puma.js:422:12
panZoomDelayed event index.html:814:11
2021-10-07 04:19:03 - heartbeat puma.js:112:12
10/7/2021, 4:19:03 AM
panZoomDelayed event index.html:814:11
UN-SUBSCRIBE rtbpm_l01_05_x puma.js:454:12
UN-SUBSCRIBE rtbpm_l02_01_x puma.js:454:12
UN-SUBSCRIBE rtbpm_bc01_03_x puma.js:454:12
SUBSCRIBE cbpm_sfel01_02_x puma.js:422:12
panZoomDelayed event index.html:814:11
UN-SUBSCRIBE rtbpm_bc01_04_x puma.js:454:12
SUBSCRIBE rtbpm_bc01_05_x puma.js:454:12
SUBSCRIBE rtbpm_l02_02_x puma.js:454:12
SUBSCRIBE cbpm_sfel01_03_x puma.js:422:12
UN-SUBSCRIBE rtbpm_l02_03_x puma.js:454:12
UN-SUBSCRIBE rtbpm_l03_01_x puma.js:454:12
panZoomDelayed event index.html:814:11
SUBSCRIBE rtbpm_l03_01_x puma.js:422:12
panZoomDelayed event index.html:814:11
UN-SUBSCRIBE rtbpm_l03_01_x puma.js:454:12
  
```

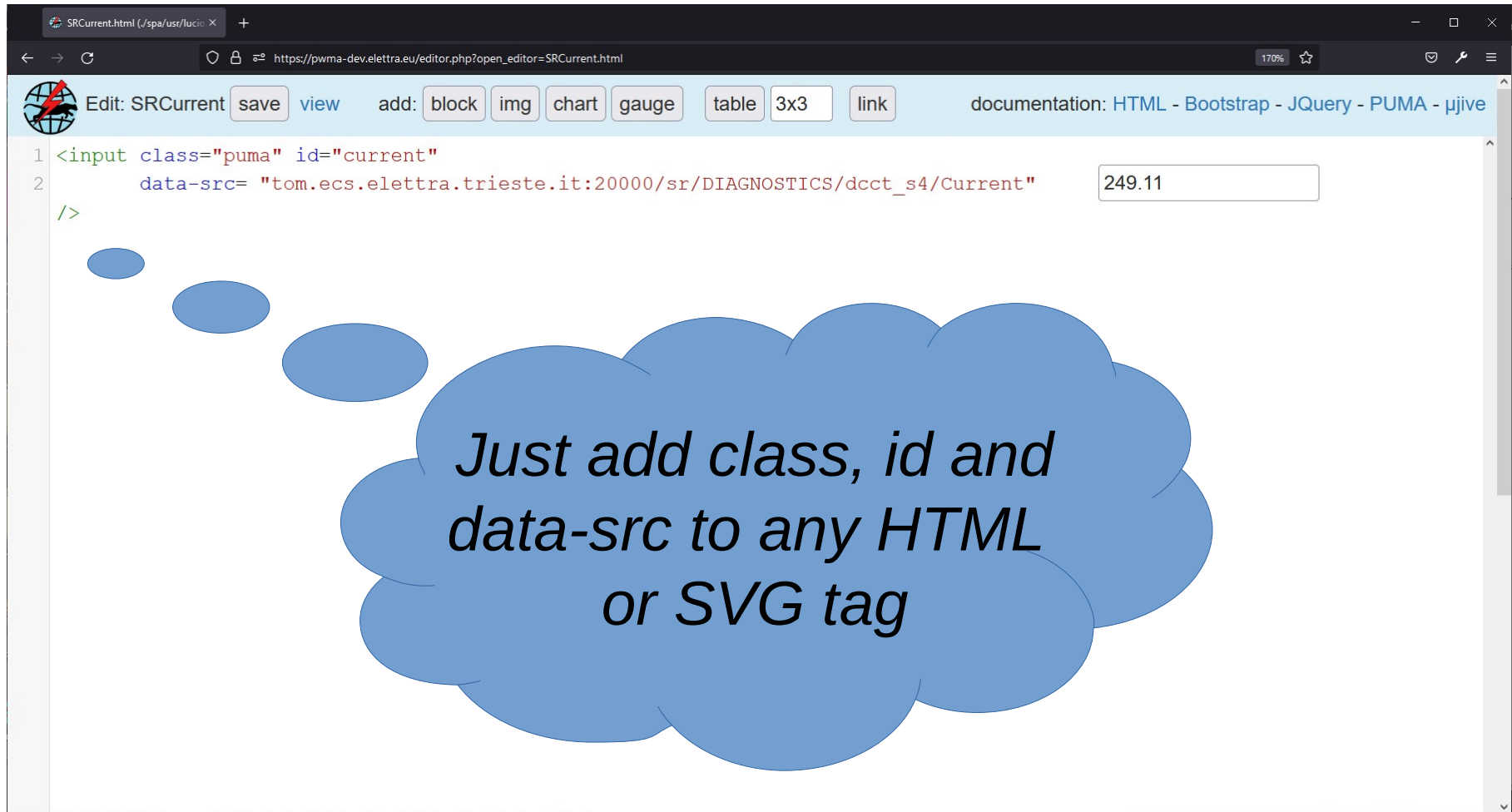
[youtube.com/watch?v=z7FUDB7w2aw](https://www.youtube.com/watch?v=z7FUDB7w2aw)



3. DESIGNER

- The designer produces *responsive* pages using flexbox which is CSS3 standard
- pages are saved in JSON format in a DB
- JSON files are interpreted by a web page and by an app
- saved pages can be used as modules embedded in new pages
- Advanced users can use a web HTML text editor integrated with an instant preview (triggered by keyup event)





SRCurrent.html (/spa/uss/Lucio) +
https://pwma-dev.elettra.eu/editor.php?open_editor=SRCurrent.html 170% ☆

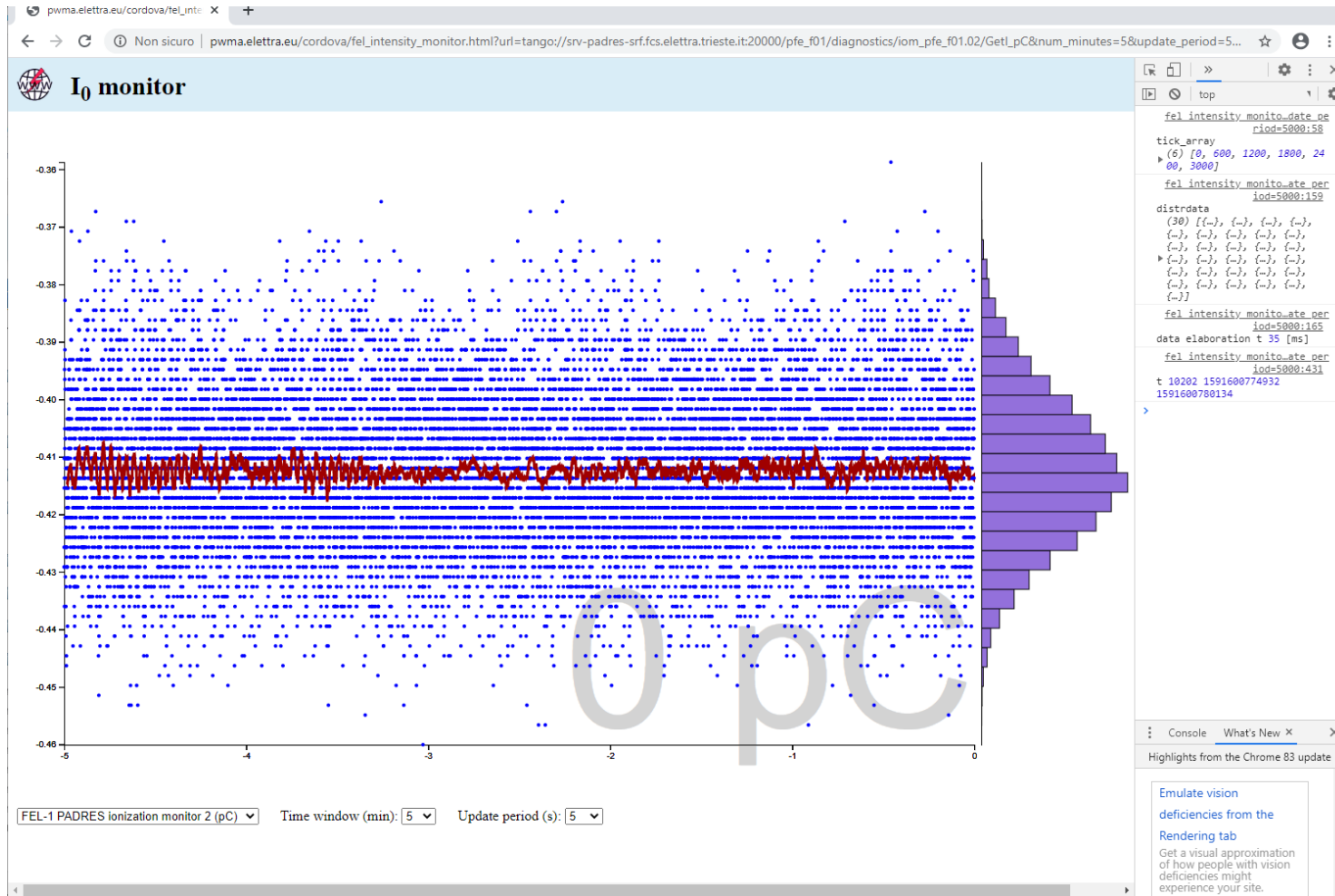
Edit: SRCurrent save view add: block img chart gauge table 3x3 link documentation: HTML - Bootstrap - JQuery - PUMA - mjive

```
1 <input class="puma" id="current"  
2   data-src= "tom.ecs.elettra.trieste.it:20000/sr/DIAGNOSTICS/dcct_s4/Current" 249.11  
>
```

Just add class, id and data-src to any HTML or SVG tag

youtube.com/watch?v=psj4ZOz7ThA



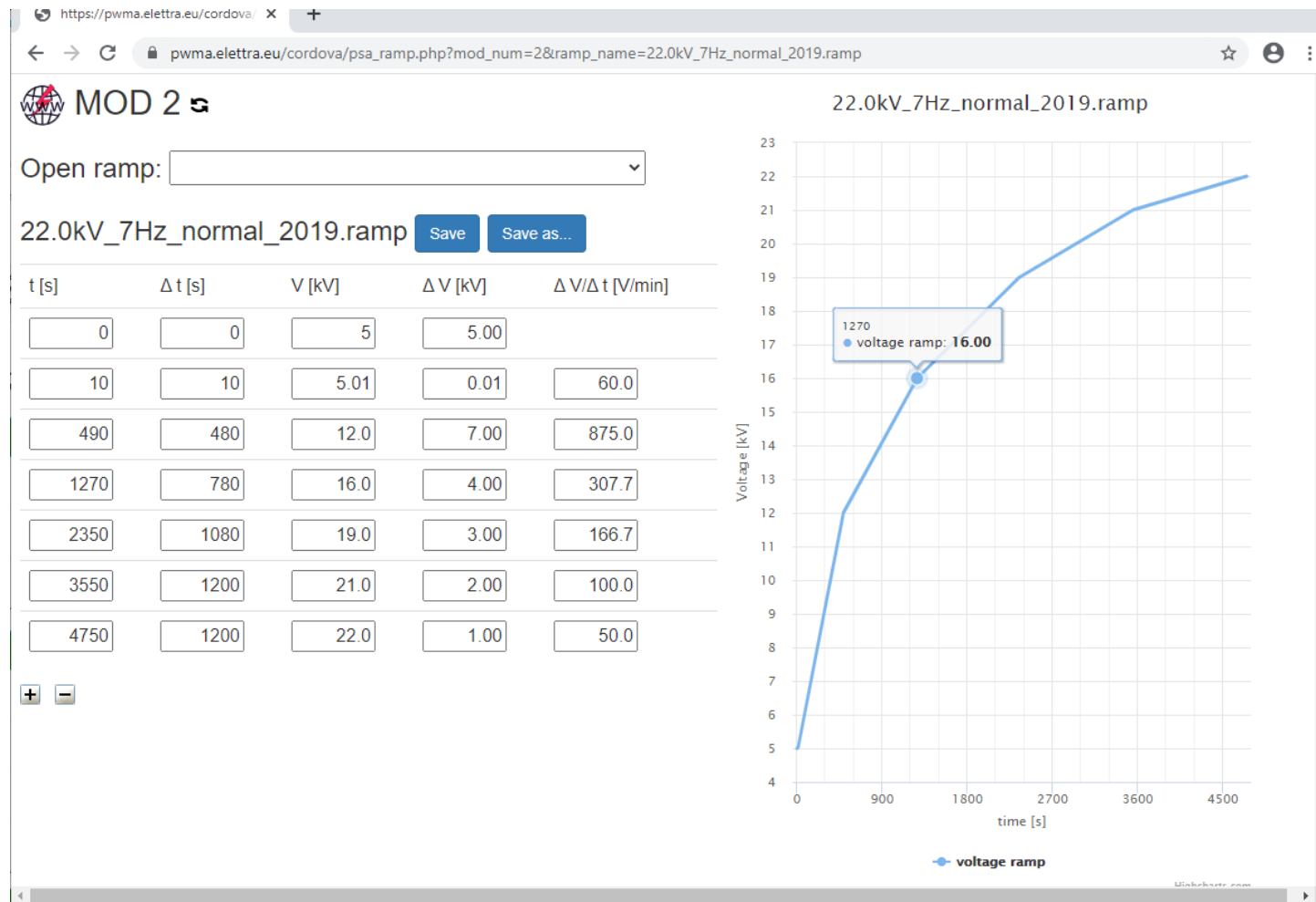


youtube.com/watch?v=gEQRNfXmVG8





WEB INTERFACES



[youtube.com/watch?v=9TI2S3mThmQ](https://www.youtube.com/watch?v=9TI2S3mThmQ)





Elettra
Sincrotrone
Trieste

Thank you!



Elettra
Sincrotrone
Trieste



www.elettra.eu