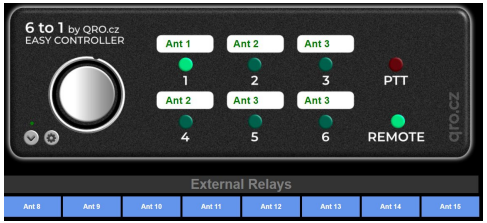
remoteSwitchFramework

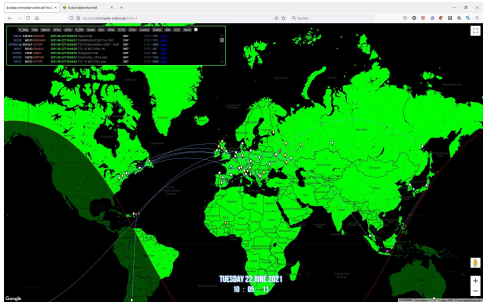
most flexible web based architecture approach

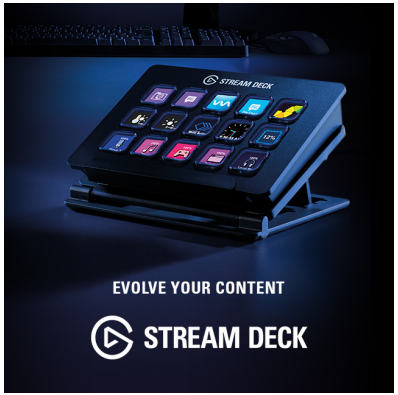
Flexibility counts

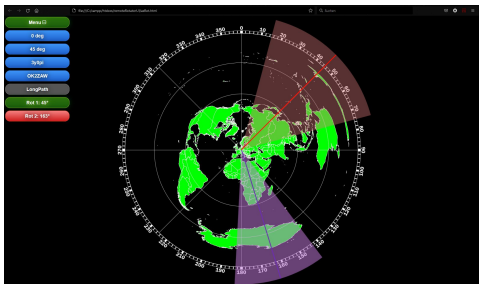
- don’t overload the mcu with running a full webserver + serving all webfiles to the client. A mcu is a mcu and not a webserver. Don’t interrupt the process of switching caused by webserver tasks delays.
- don’t story any kind of businesslogic besides the process of switching on the mcu.
- follow important software architecture and design principles: seperate hardware from process logic, seperate process logic from presentation/ui
- keep the business logic under our control, so there is no need to upload the firmware of the mcu due to changes in the (webstored) business logic. Easy bug fixing, easy updates!
- provide a default set for serving the web files from qro.cz, but be most flexible where to re-route the files to any location needed.
- provide a default set of web pages for download/zip to be modified and stored to a (local) webserver
- new: provide a desktop app to spawn a webserver in the background. A user does not need to install a webserver
- Cave eat: Solve the problems due to browser compatibility and security. Since this system is complex and support intensive, all files and firmware are copyright protected. This is not open source, but you as a customer of qro.cz will have all benefits using it.
- All qro.cz/remoteSwitch devices implement the same API for http and websockets. So it’s easy to implement it for custom User interfaces, node red, Touch Portal or stream deck, your own Contest-Dashboard or mobile apps or or or…

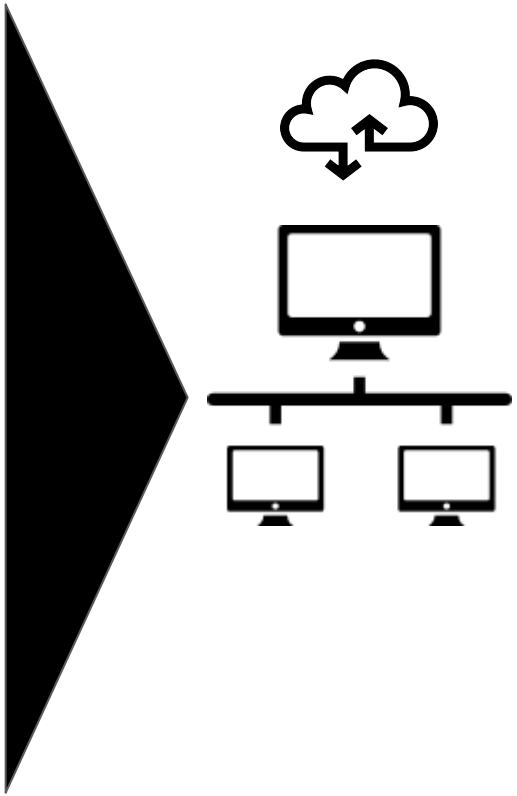
Decouple UI from hardware = most flexible, always up to date









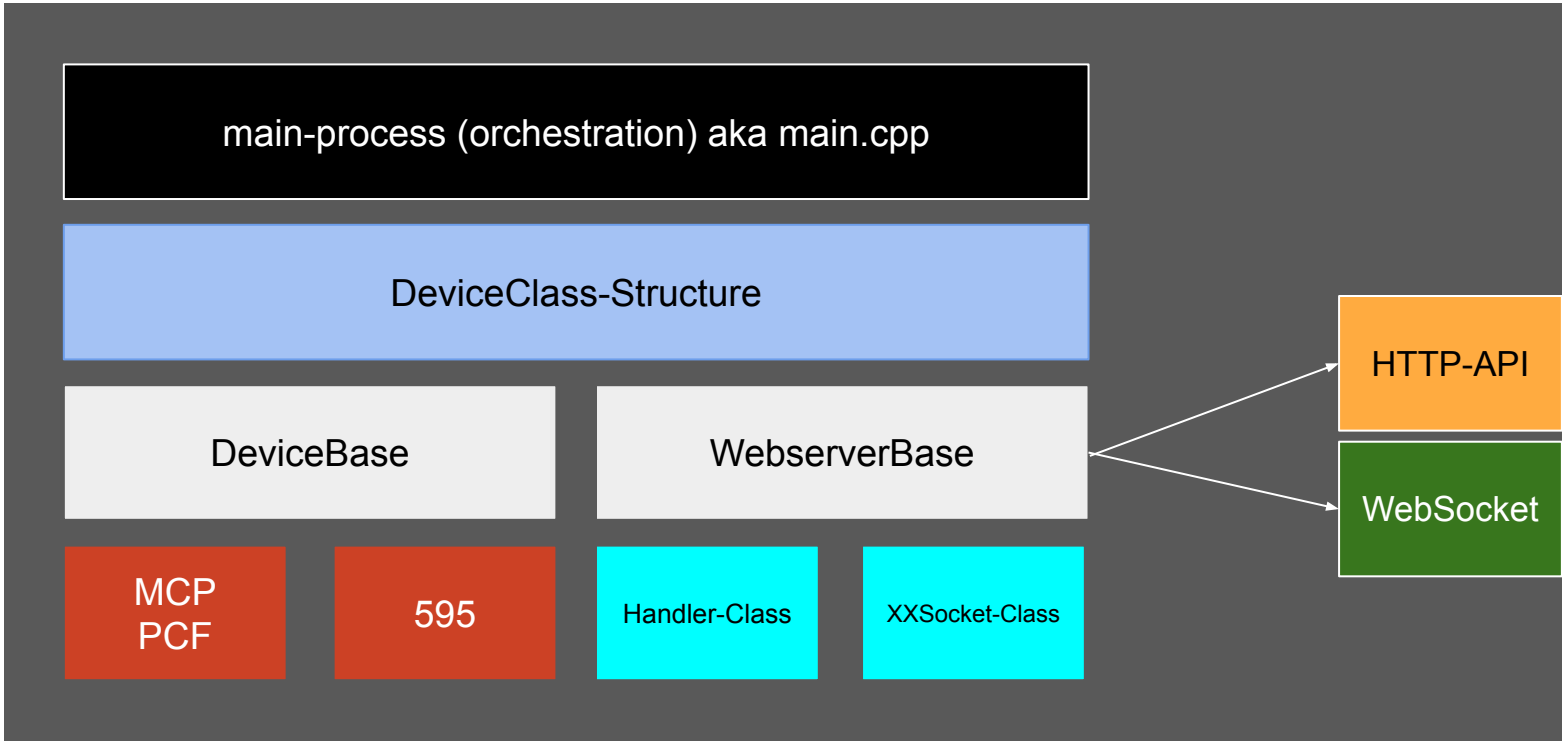


controller

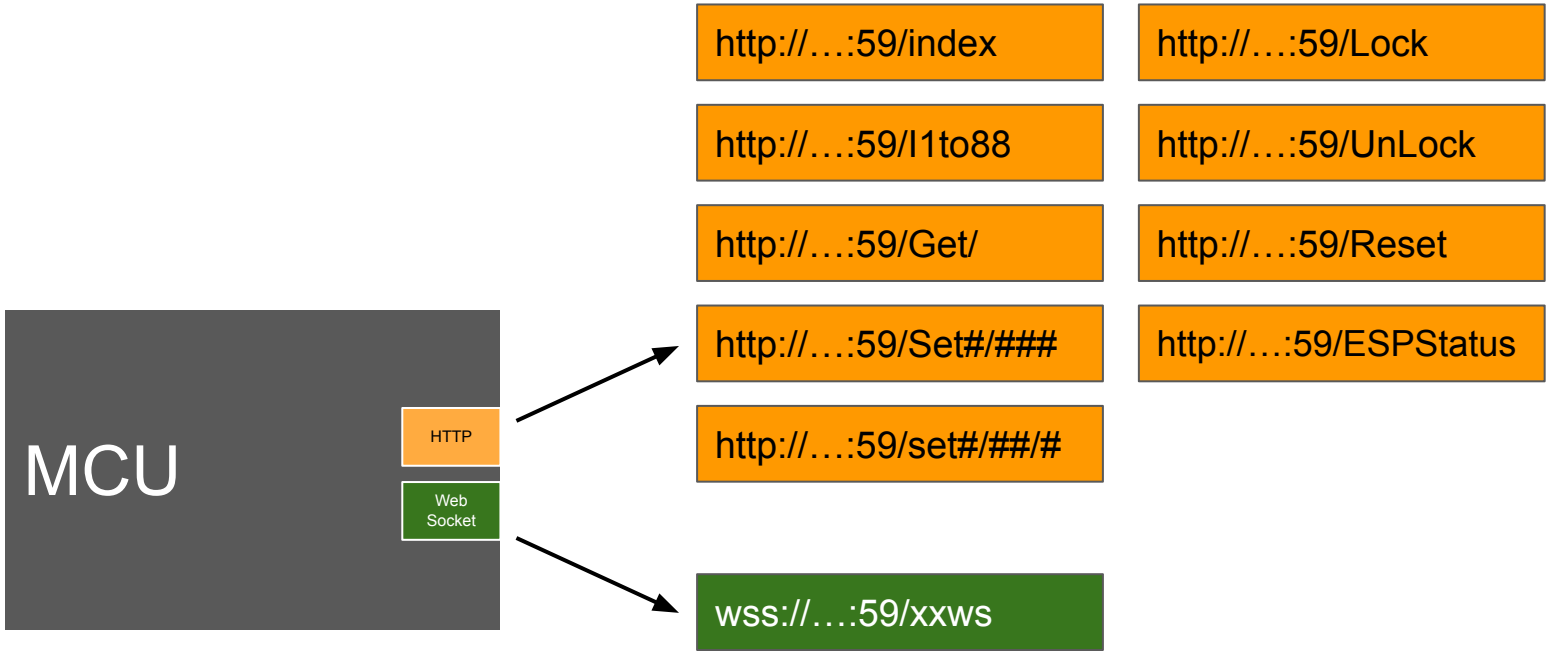
controller

UI can be used for:
SM1-2
SM1-3
4to1
6to1
6by2
10by2
K9AY
4WAY
4SQ
4SQ-8Dir
...

Architecture in C++



Web API: http&ws



Simple Web API commands

http://<ip>:59…

/Get/
/Set<banknr>/<bitvalue>
/set<banknr>/<pinnr>/<0/1>
/Lock
/UnLock
/Reset
/ESPStatus

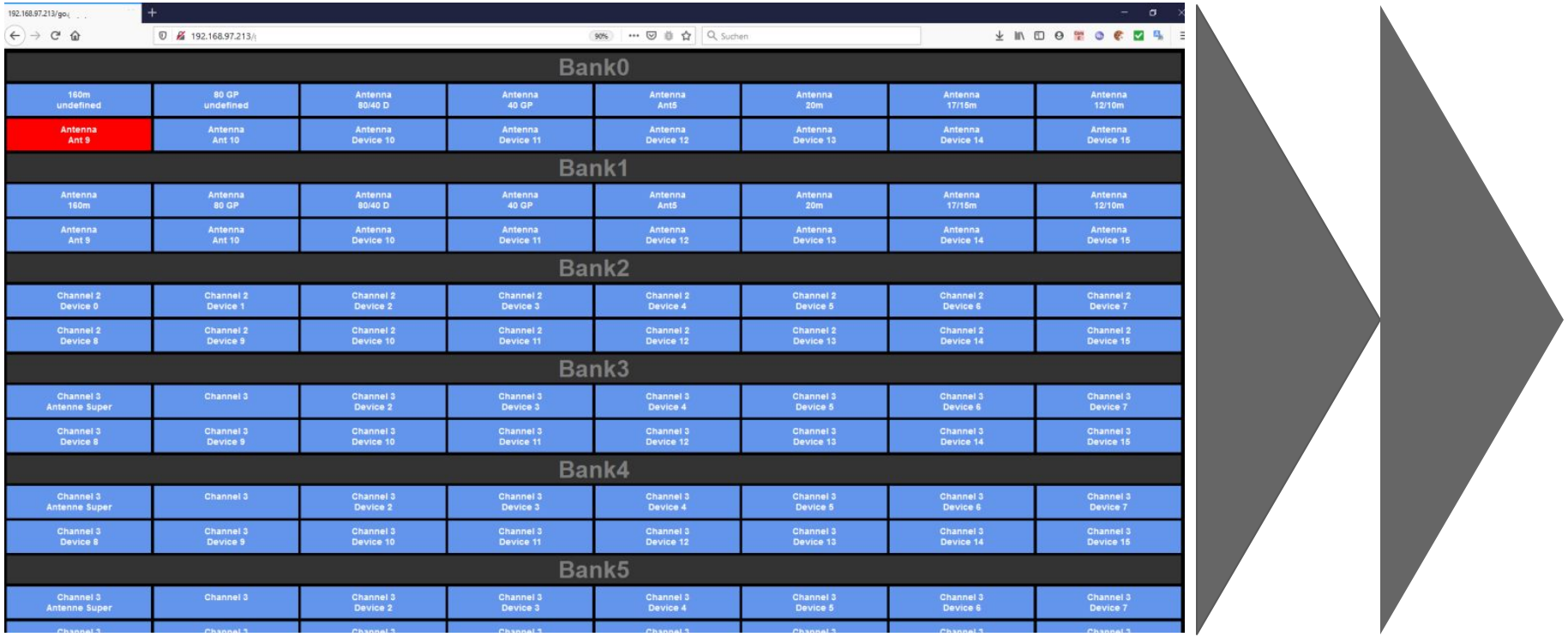
ws://<ip>:59/xxws

G
S/<banknr>/<bitvalue>
s/<banknr>/<pinnr>/<0/1>
L/<0/1>
R

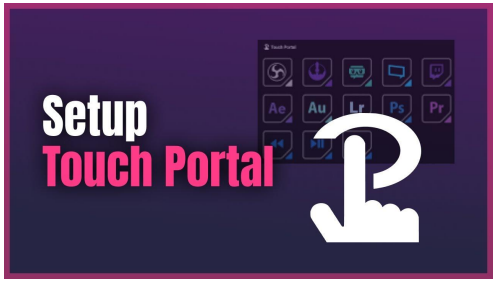
Special implementation 6by2: S/s is X/x
X/<banknr>/<bitvalue>/<radionr>
s/<banknr>/<pinnr>/<0/1>/<radionr>

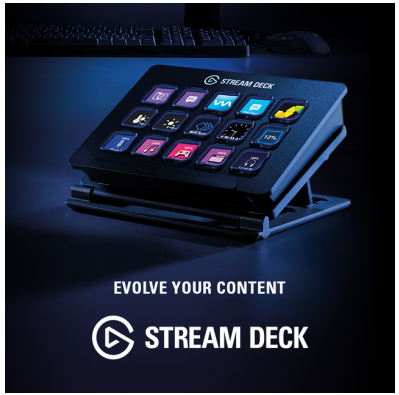
Status requests and updates can be done by using http and/or websocket api.
Using the defaultUI, as configured in your custom config (json) file.

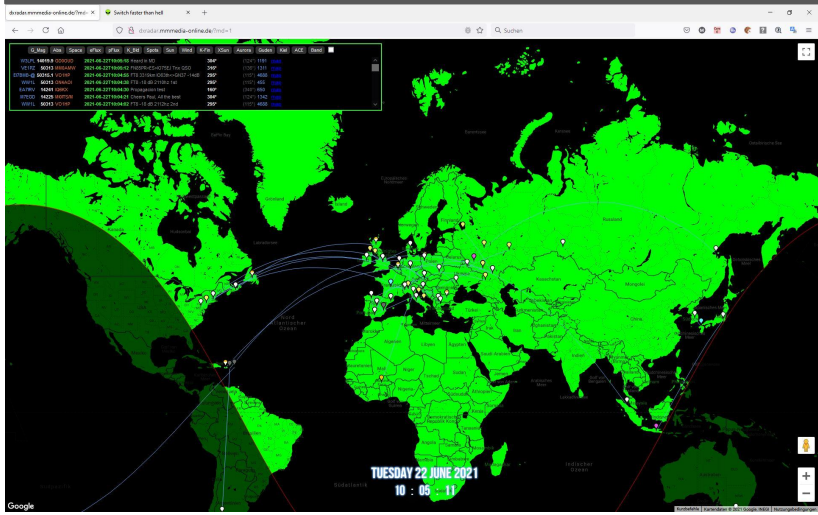
Using defaultUI

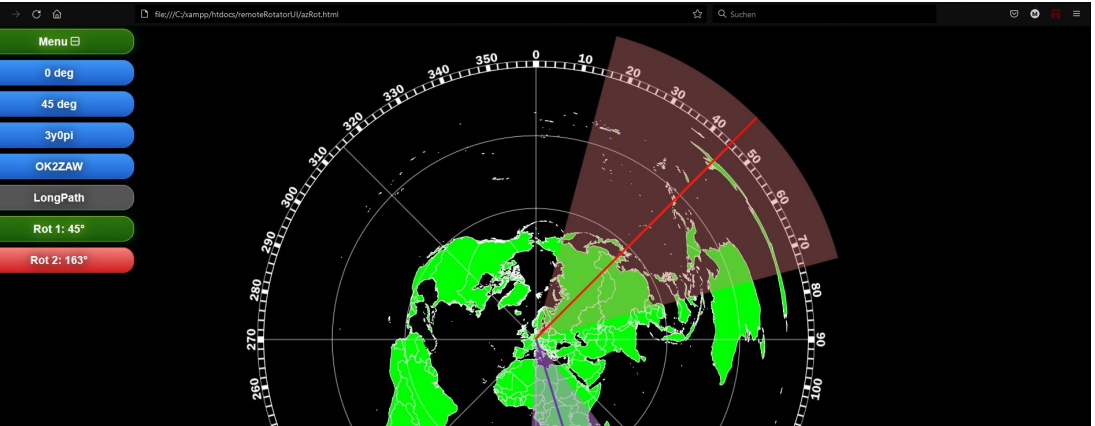


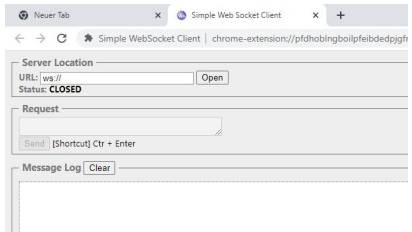
Using Web API

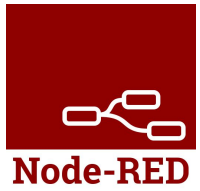




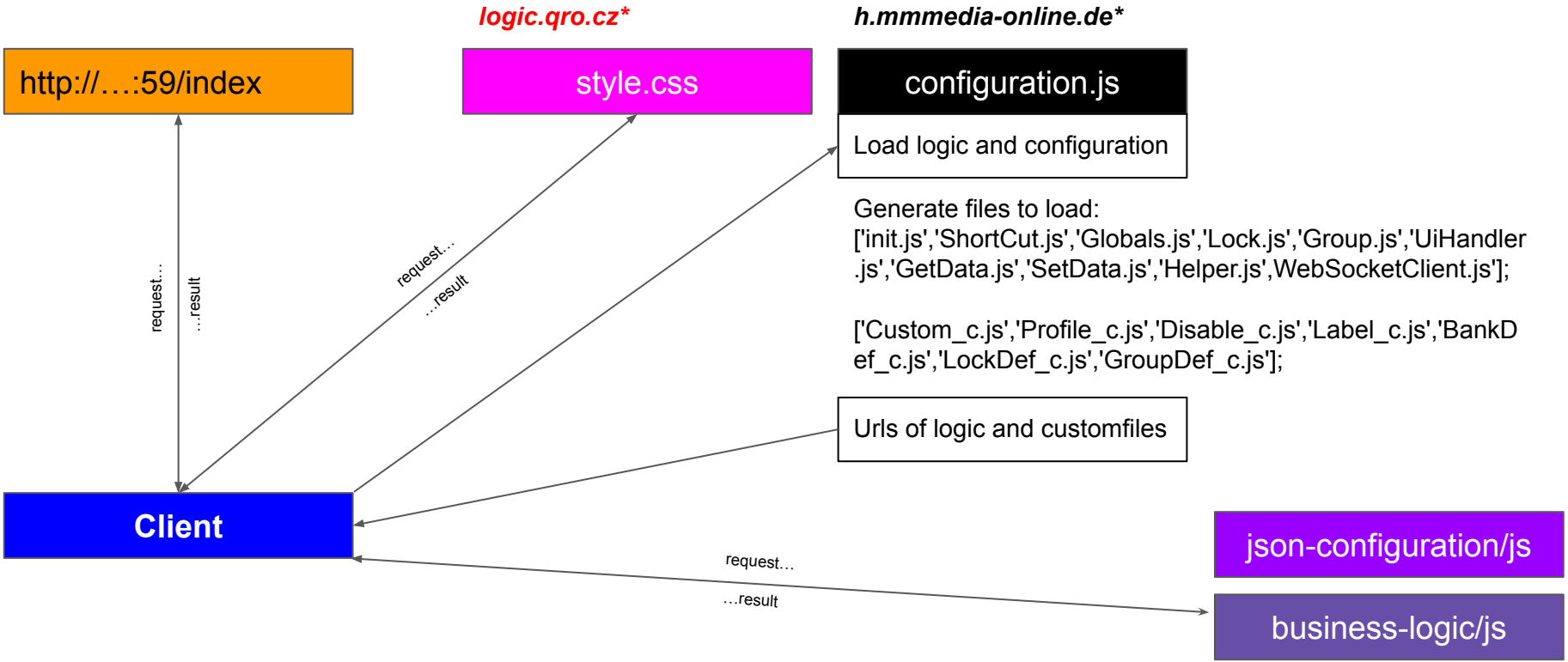








Using defaultUI: …:59/index



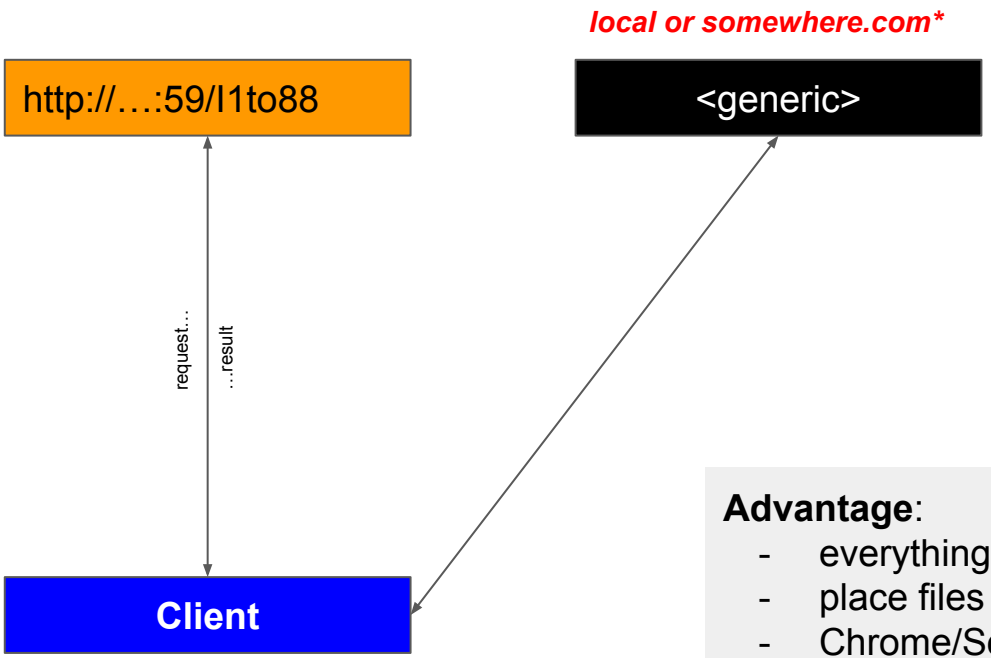
default can be changed by paramaters*

fixed path to configuration.js*

Using defaultUI: …:59/index?<*parameters*>

t	<typeidentifier> for dominator (/l1to88)
ptd	url to full generic generator (/lto88)
css	url to css file (configuration.js)
redirect	redirect to specified url (configuration.js)
ipofswitch	(local) ip or url of the switch (configuration.js) (without protocol)
portofswitch	port 59 as default firmware port (configuration.js)
jsurl	url where all the js files are located (configuration.js) (without protocol)
customizeurl	url where all config json files are located (configuration.js) (without protocol)
nrofboards	number of 16ch banks (only caller.html, see “files on local computer”)

Using defaultUI: …:59/l1to88



Purpose:
<generic>-Page (js or php, etc.) is responsible to inject the urls for the needed businesslogic and custom-json files.

Also the url for css is generated and additional functions called (like init() in onload())

Advantage:
- everything you want can be injected.
- place files wherever you want (local/sever)
- Chrome/Security problems solved since “switchip” serving the html page

Using defaultUI: …:59/l1to88?<*parameters*>

t	<typeidentifier> for generic generator like dominator.php (/l1to88)
ptd	url to full generic generator (/l1to88) (incl. protocol)

\+ all parameters of the generated urls to js-files

Using l1to88 solves the issue, that all pages should be served by the webserver, in this case the impero (Cave eat). Issues with access private network can be solved. And this is the most flexible way of page generation by javascript. All is dynamic :)

Restrictions and know issues

CORS - Cross origin requests
Affected: All browsers. Chrome vy restrictive.
Solution: Add header in firmware (done).
Use /l1to88, App, VPN, Firefox or Cors-Browser-Plugin if running into issues

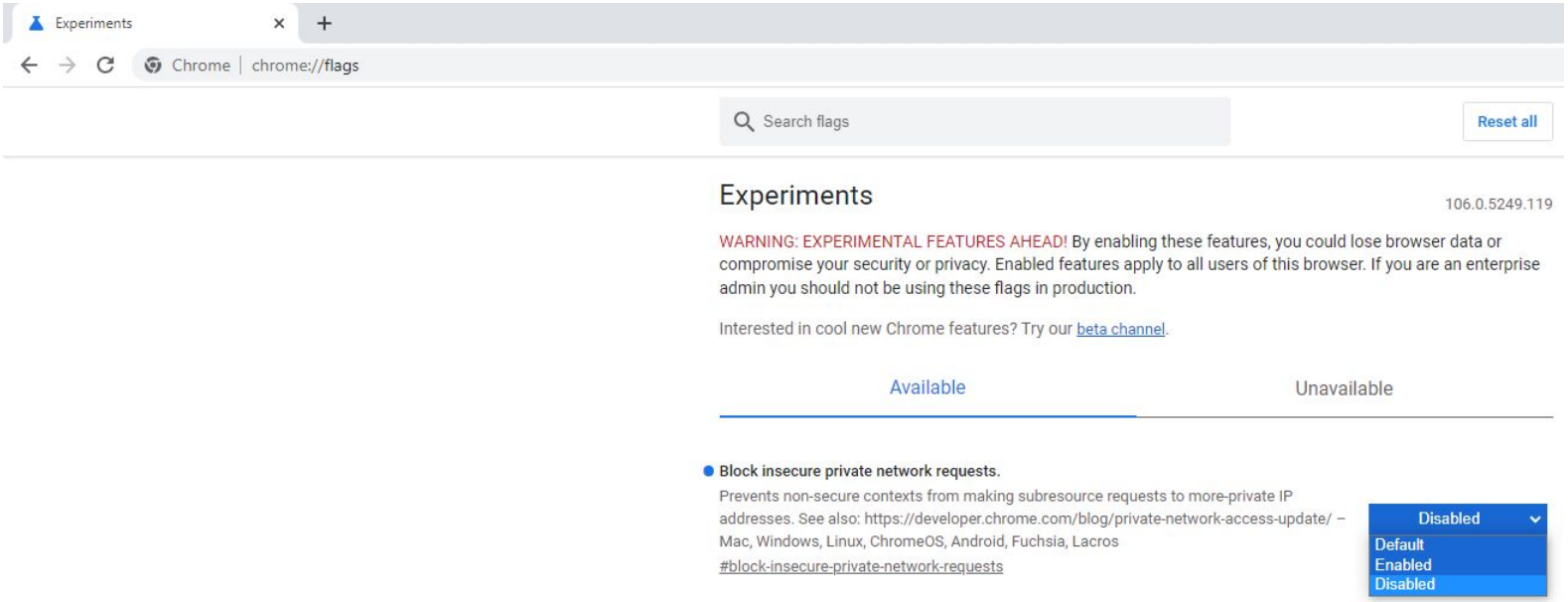
APPLE IOS or OSX
Officially not supported since there is no OSX device available. DefaultUI might work since it was tested on IPad, but more complex special UI for qro-devices using Web API might not. Get in touch in case of issues.

Supported browser is Firefox
Since this is a real complex Solution, focus was on one browser with best platform compatibility. Chrome might work, too, but not for the pinger.html for example... Please report bugs to remoteswitch(at)gmx.de

https
A small mcu is unfortunately in short of memory. The certificates and encryption takes a lot of performance. So using a VPN is really recommended in general anyway, all traffic is http. Win-win-win for all. There might be more an more issues with browser restriction, so using a https to http gateway might be useful one day. We keep our eye on that issue!

Restrictions and know issues

Upcoming: F*ing Chrome implementation of RFC preventing CSRF**
Chrome decided to go his own way with private network access. Using the /l1to88 system will help here, but atm you still can do this if you are forced to use chrome (maybe older portable versions will help,too):



Option (A) Using Files on your local Computer

Option (D) Using your own webserver (local/public)