PAUL SCHERRER INSTITUT

PSI

Simon Rees :: Software Engineer :: Large Facility Controls Section

Control System GUI Development @ PSI

Controls GUI and Centralised Application Platforms Workshop September 2022

Introduction

At PSI we are mostly still using traditional (non-web) technologies...

But a few of us have been working on pilot projects for the web

Some colleagues support this; others view the applicability of the web in the context of our lab with deep skepticism !

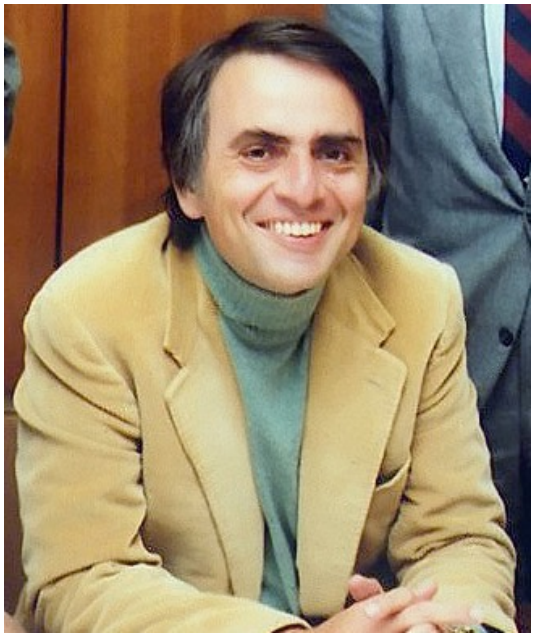
In this talk I will focus on the realities of the situation in our lab and my experiences with developing and deploying a new software solution (called “WICA”) which aims to bring our control system to the web.

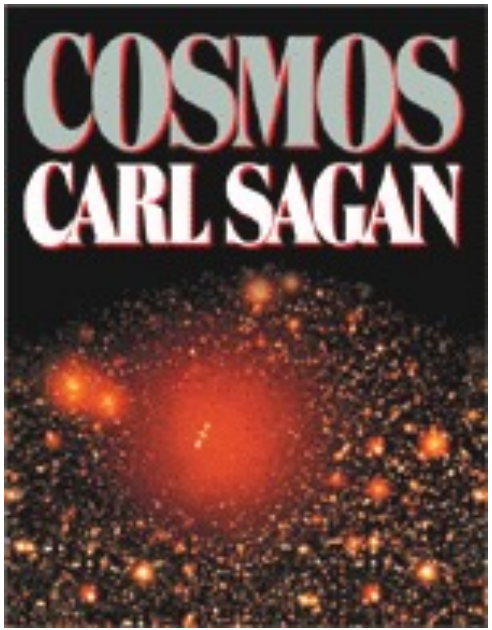
PAUL SCHERRER INSTITUT

Traditional approaches to GUI development

GUI Development at PSI – some context…

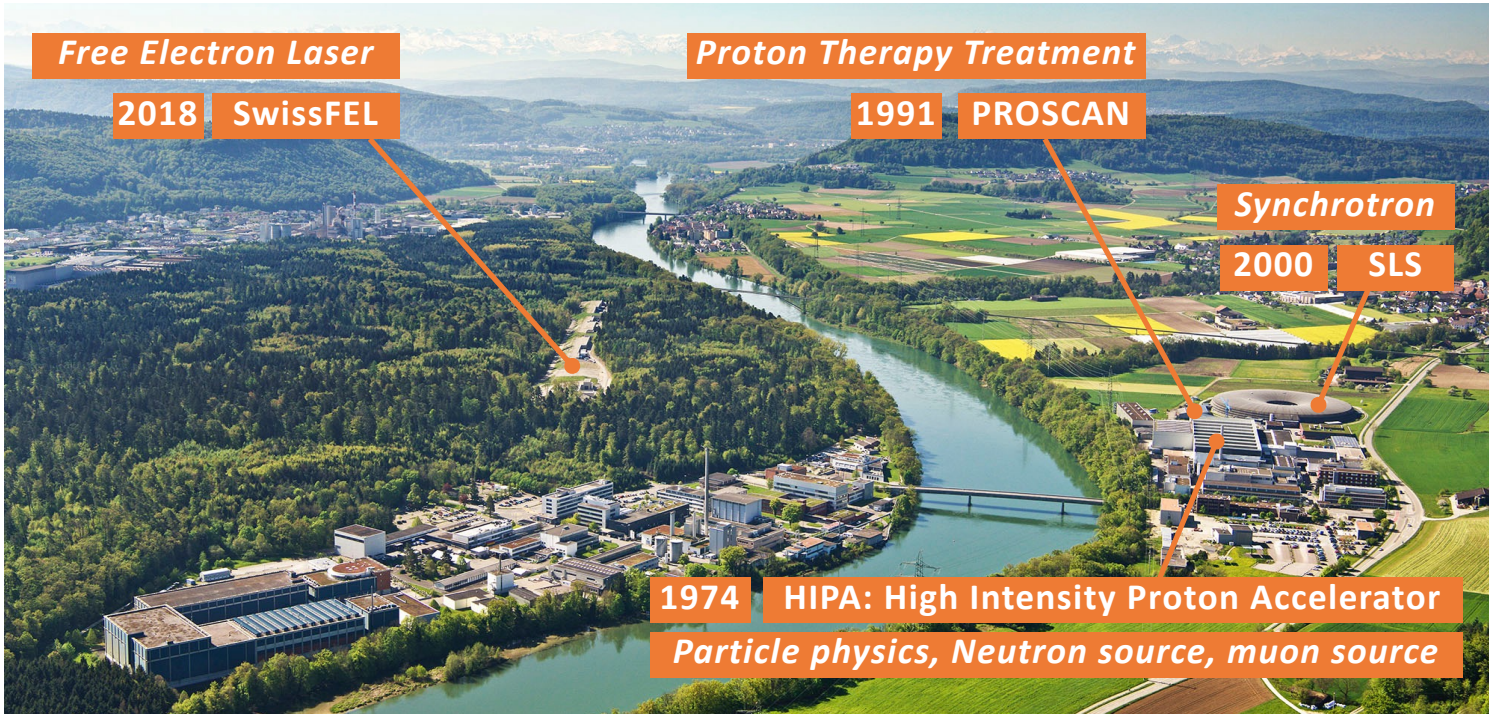
"*You have to know the past to understand the present*" - Carl Sagan 1980





By credit NASA JPL - https://commons.wikimedia.org/wiki/File:Planetary_society.jpg, Public Domain, https://commons.wikimedia.org/w/index.php?curid=24462106,3295528

Facility Overview



We are a group of ~**30** people with **4** accelerators to support
They are used for science research and medical treatments

We support multiple generations of control system technology
We are very busy and feel very resource constrained !

PSI Controls – Services & Stakeholders

We provide

- the **control system** for each of the running machines
- **technical support** and **development** for the machine stakeholders

We have > **60** stakeholder groups

- Machine Operators – who *run our accelerators on a daily basis*
- Technical Specialists - who *support the magnets, RF, vacuum systems, …*
- Scientists – who *support the scientific users and who drive forward machine enhancements*

It's difficult to work out the optimal structure for our institute

- we are now onto our third organizational structure in 6 years !
- this is particularly challenging for our control system architectures !

"Any organisation that designs a system will produce a copy of the organisation's own communication structure" – Conway's Law

Our future architectures will need to be more flexible

- if we move to less monolithic, finer-grained systems then responsibilities can be reassigned more easily and different components can evolve at different rates
- Microservices might help, but evidence now shows such architectures incur significant additional costs !

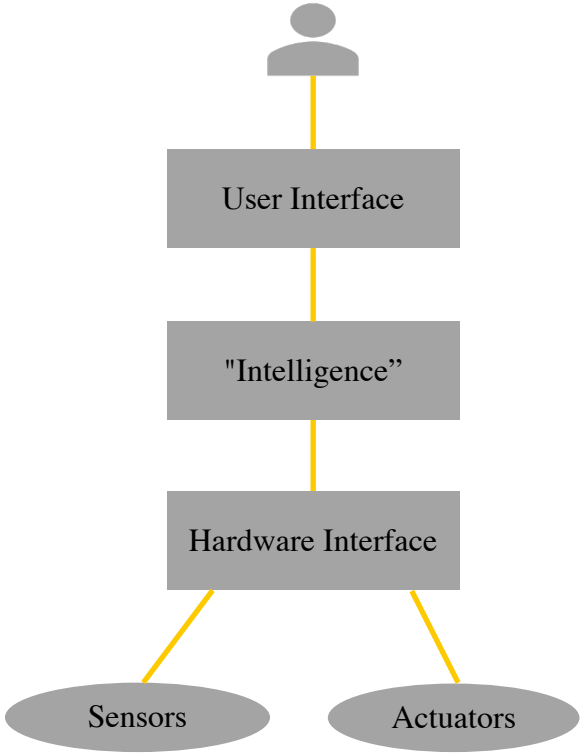
Use of EPICS

- Previously PSI had multiple in-house control systems
- Since 2009 we standardised on EPICS
- Most of us agree this was a good idea… ☺

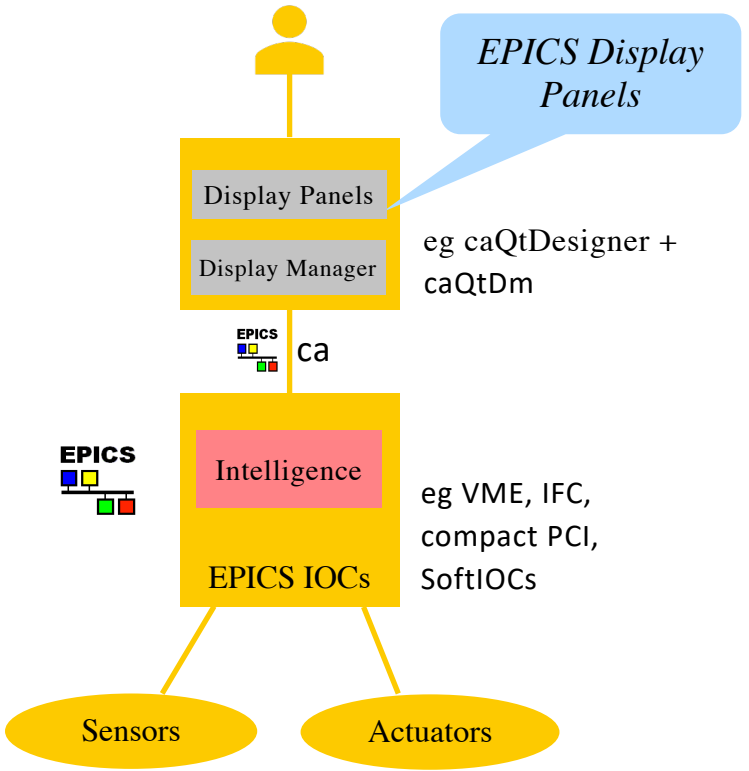
…but, we can leverage EPICS in many different ways…

…and we don't always agree the best approach…

The Control System Technical Stack



EPICS integration - Approach 1



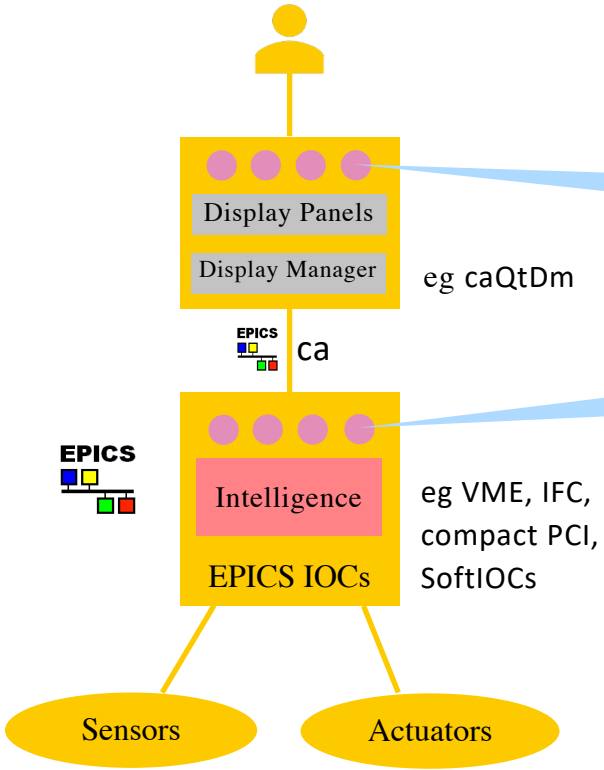
"Use EPICS for the entire technology stack"

Yay, with this design we can sometimes realise the dream of getting our users to create their own user interfaces... ☺

But... they are sometimes more like *data* interfaces than *user* interfaces ! ☹

EPICS integration – *data* interfaces vs *user* interfaces

The central abstraction of EPICS is a control point, or *channel*



User Interface
points-of-interest

Control System
channels

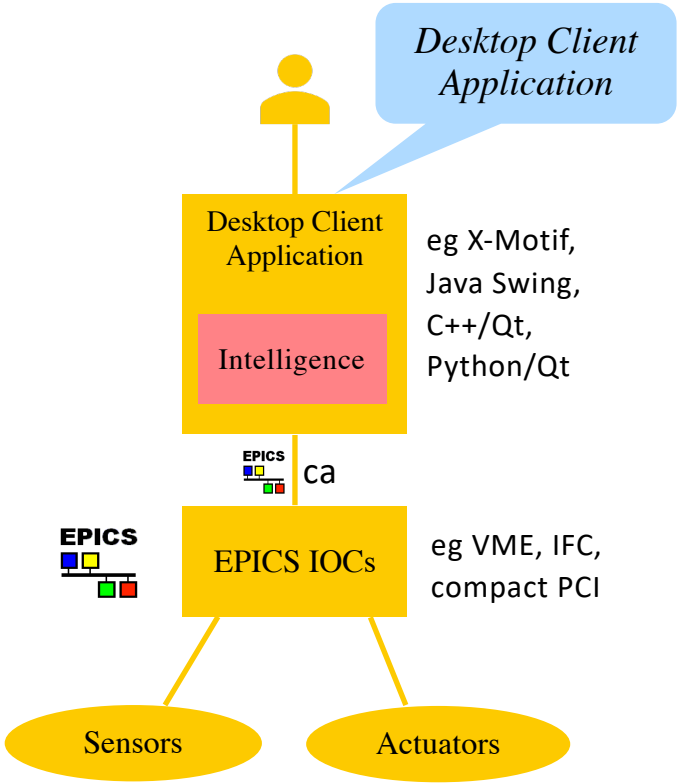
1 -1 mapping

What's perhaps missing from this design is the GUI "intelligence" (= state manager) to help guide the users in their workflows

For *trained operators* this may be fine !

But for newbies the complexity can be frightening !

EPICS integration – Approach 2



"use EPICS as a flexible field bus, but use other technologies on top..."

The user experience with a desktop client applications is potentially (but not always) richer ☺

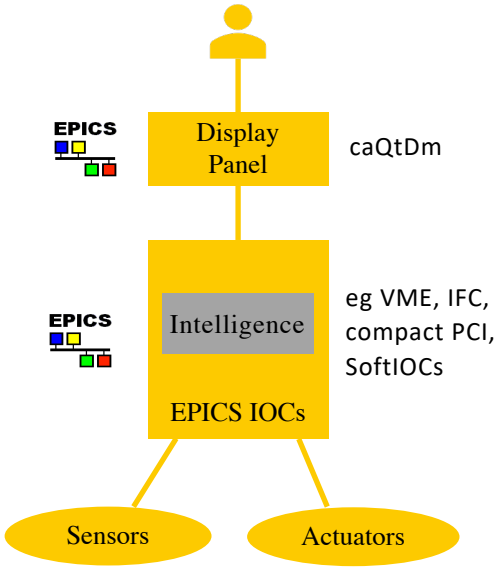
But it requires the ~~enslavement~~ *services* of a control system engineer ! ☹

(ideally one who cares about UX !)

Classic GUI Architecture at PSI

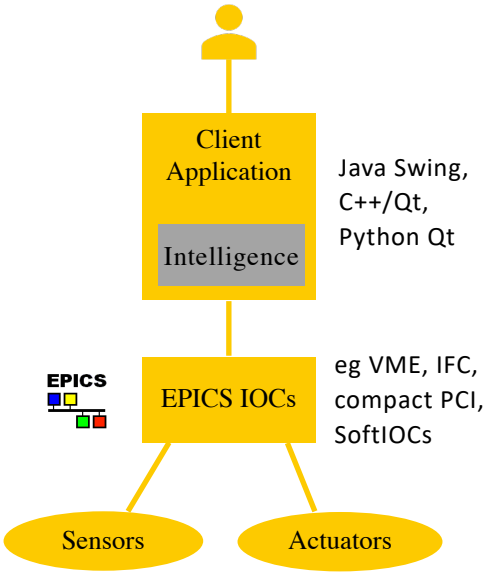
"*You have to know the past to understand the present*" - Carl Sagan 1980

SLS and SwissFEL were designed from the start to use EPICS



Fat IOC / Thin GUI Client

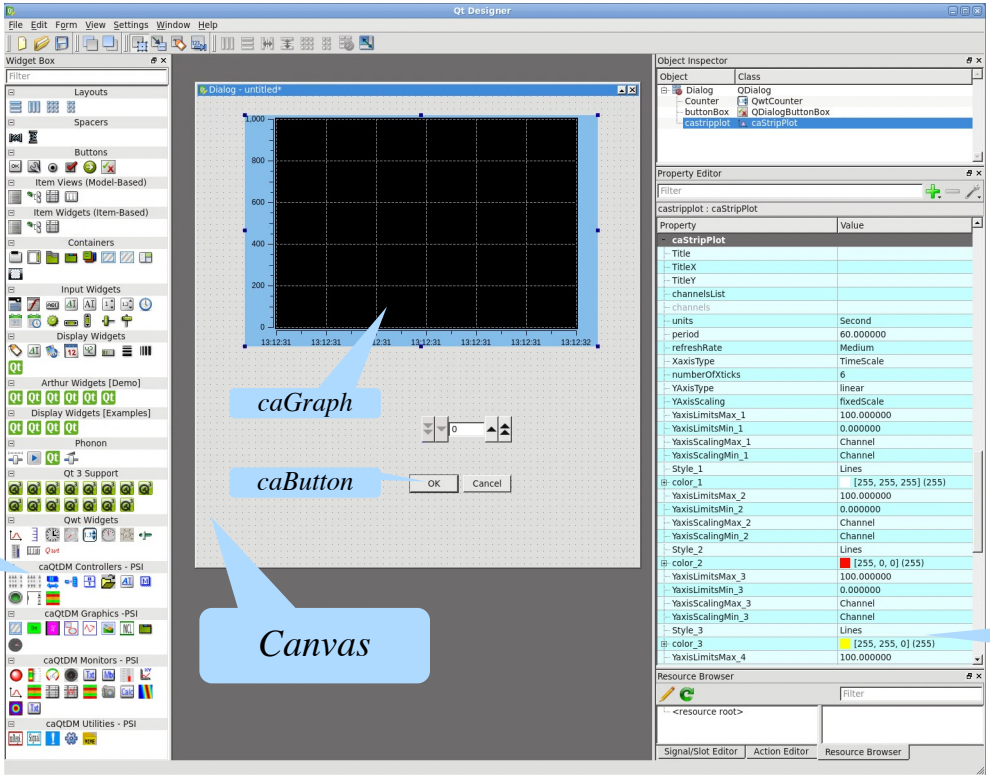
HIPA and PROSCAN started with a different control system architecture (ACS) and were migrated to EPICS later



Thin IOC / Fat GUI Client

=> our GUI architectures of today are determined by events > 20 years ago !

PSI Display Panels – how they are made



caQtDesigner

caQtDM

Credit: Dr Anton Christian Mezger

The “gold standard” against which any new tool offerings will be judged !

Good enough in [x] % of use cases

[*] insert personal opinion here !

Used by: Machine Operators, Technical Experts, Control System Engineers

PSI Client Applications – how they are made

C++/Qt

Java / Swing

Python / Qt

A good IDE !

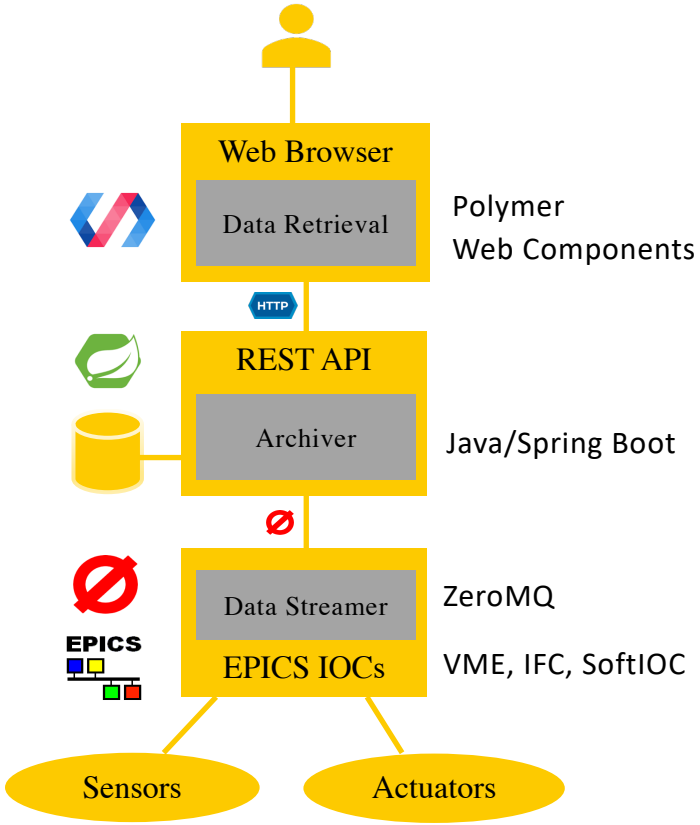
coffee !

effort !

Web-based approaches to GUI development

GUI Architectures at PSI – first web-based approach

For the new SwissFEL facility in ~2015 we started to use web-based technologies…



Use Case: Beam Synchronous Data Acquisition System

Main concepts

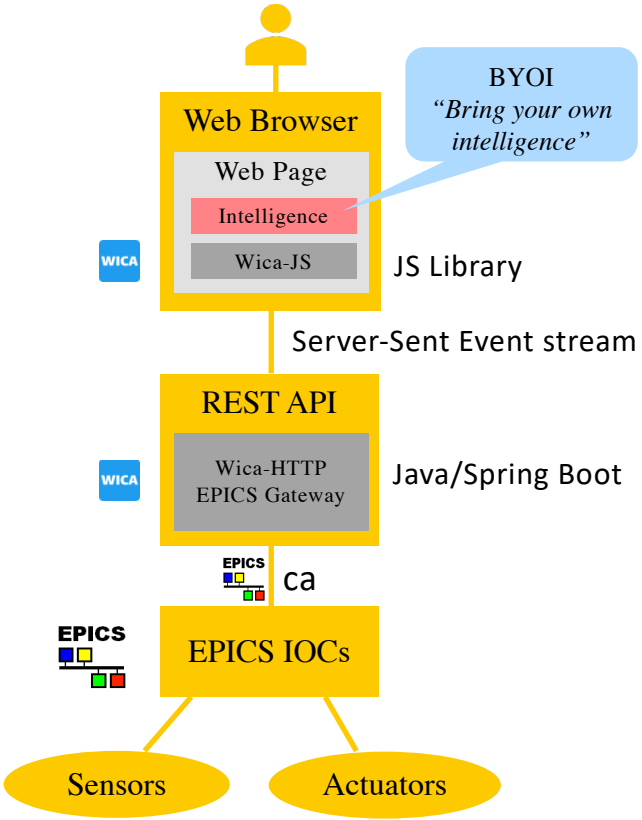
- used **ZeroMQ** (not EPICS channel access) to stream data from the IOCs
- used **Web Components** (Polymer) to provide the data retrieval user interface. No frameworks !

Learning Experiences

- it worked ! ☺
- need robust and well-defined APIs at all levels
- good monitoring is very important for fast recovery
- despite not using a heavyweight framework it was not easy to upgrade from Polymer to its successor Lit

GUI Architectures at PSI – second project - WICA

WICA = Web Interface for Controls Applications



Use Case: Facility Status Displays / Provide offsite Displays

Main concepts

- Reflect control system *points-of-interest* onto html element data-* attributes
- Use the received data to update **text fields**, **plots**, **SVG displays**, **camera images** etc onto the users display. This can be done in many ways.
- Try to provide the users with some kind of **tool** or **workflow** so they can easily create their own displays

Learning Experiences

- Partial success: WICA is the only display tool that works off-site
- Getting the data to the user's webpage is easy ☺
- Providing a tool or workflow for the users (who are not programmers) has been difficult ☹

Leveraging WICA – supported workflows

Text-only Display Page

Write some html to define your control points

WICA will render the text content of your elements directly - no JS required.

Display Page with Text and Plots

Write some html to define your control points

Choose a JS plot library eg *Plotly* or *Highcharts*

Write some JS to update the plot using the received data

Display Page with Text and SVG

Write some html to define your control points

Use an SVG Editor eg *Boxy-SVG* to create your graphic. Or create graphic with text editor

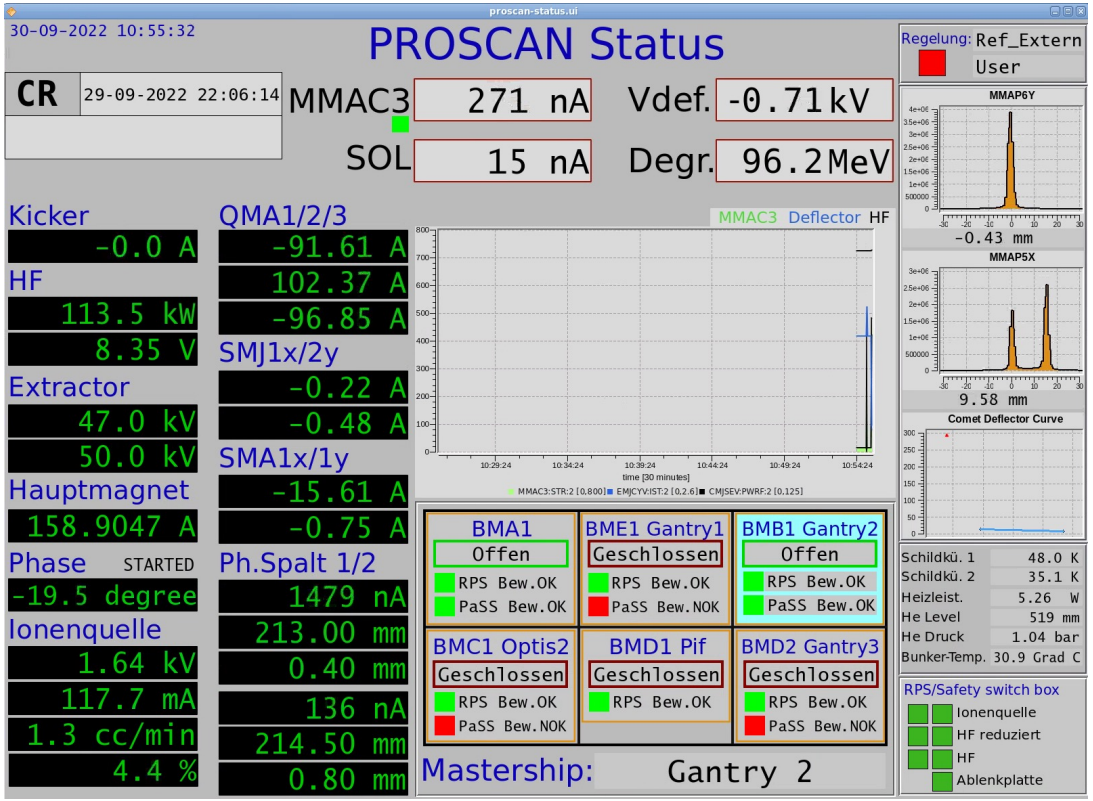
Write some JS to update the graphic using the received data

Display Page with Camera Images

Obtain a Web Component that supports rendering camera image data. At PSI we use *Lit*

Write some html to define the control points associated with each web component

Web Upgrade Journey Start - PROSCAN Status Page

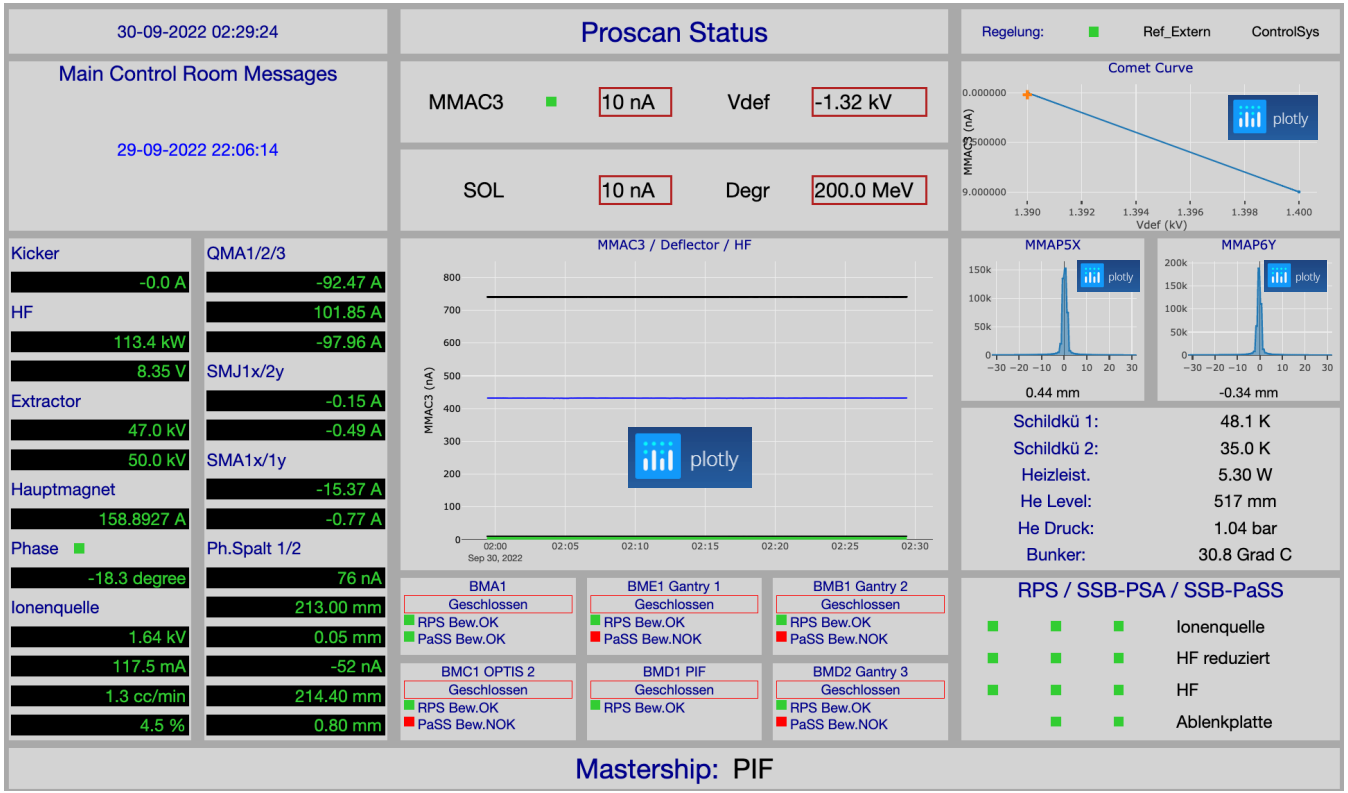


The original display page made with caQtDm

Ugly, but used effectively by many people !

Goal: make something just as ugly, but available on the web

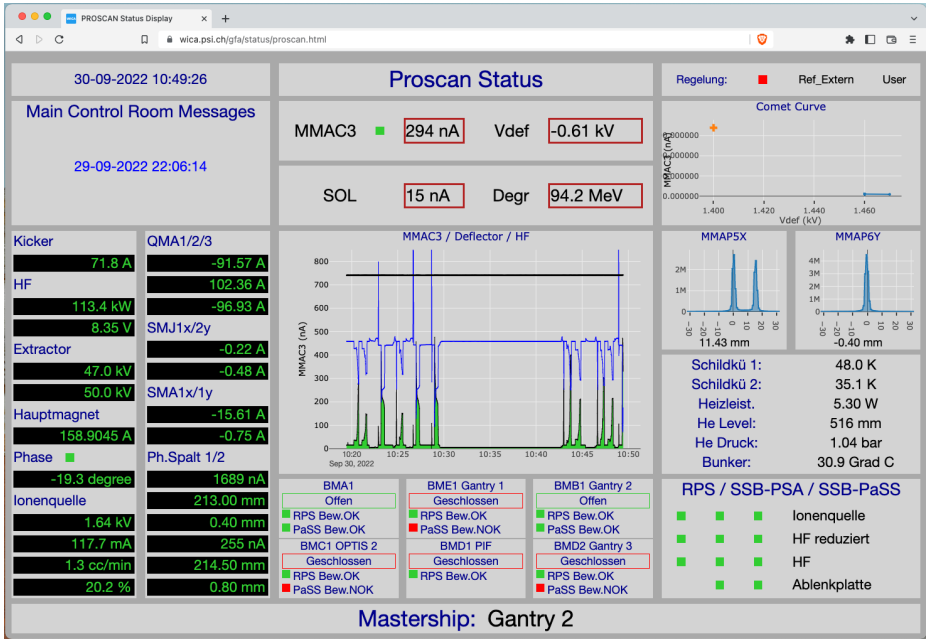
PROSCAN Status Page – Mixed Text & Plot Example



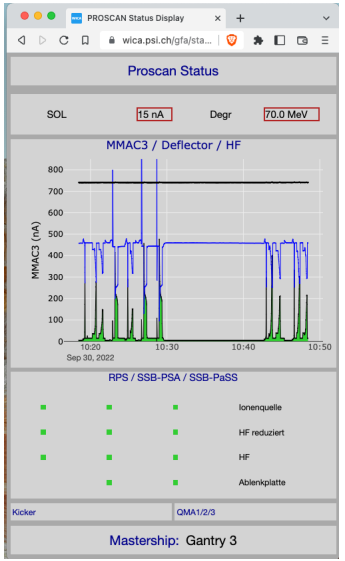
CSS3 Grid

CSS3 Flexbox

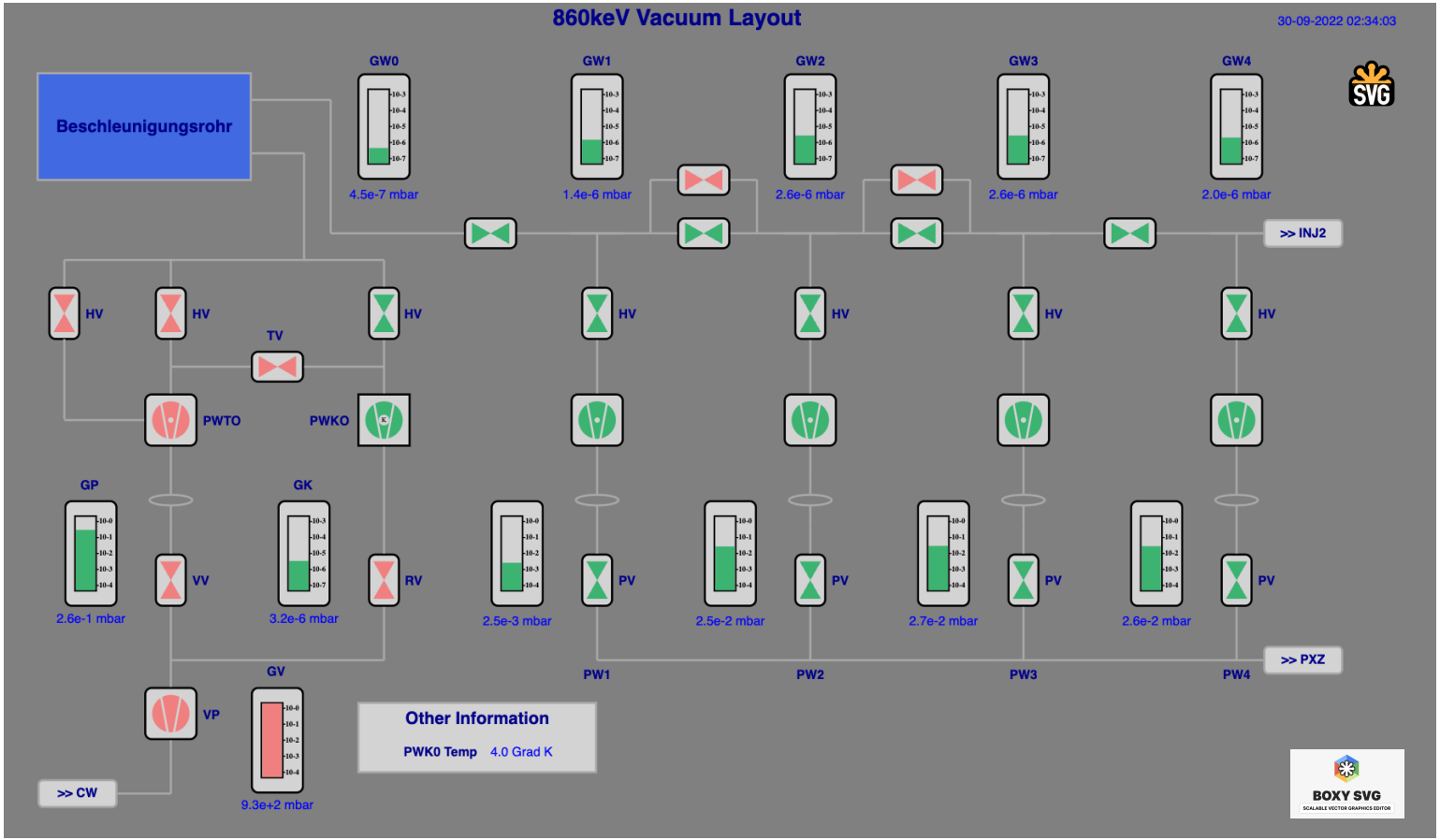
PROSCAN Status Page – responsive design



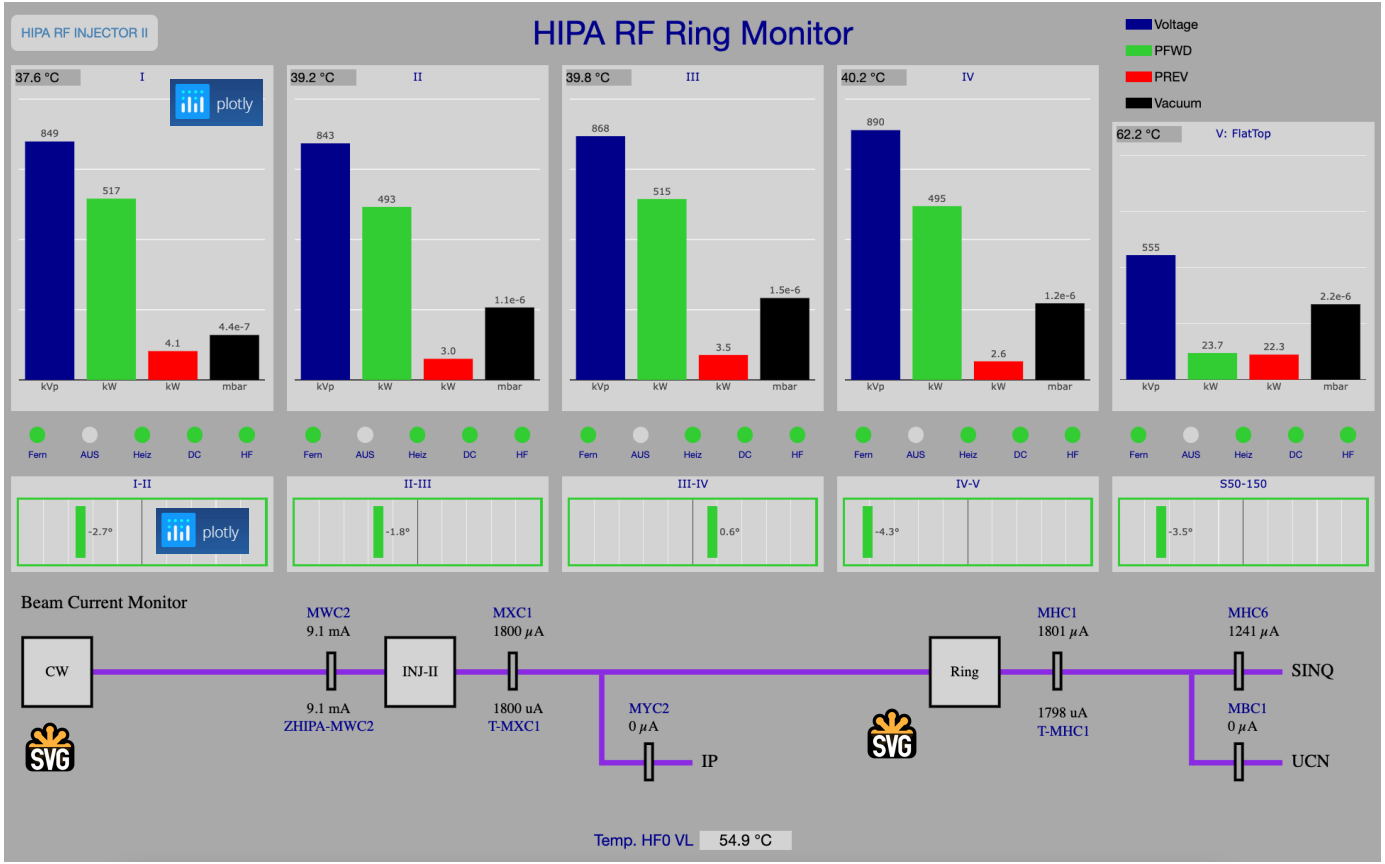
Because the pages just use normal web technology we can use CSS media queries to change the formats according to the features of the viewing device…



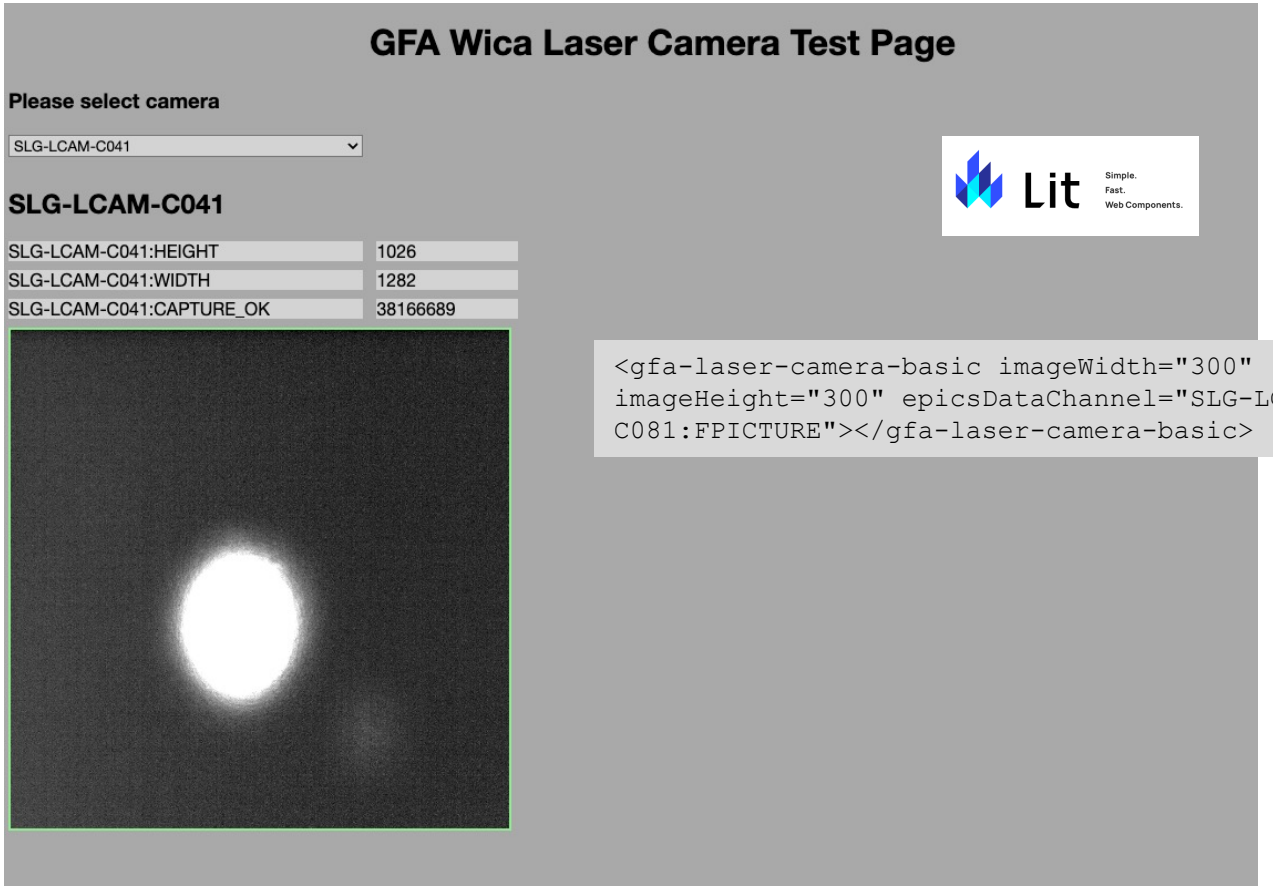
HIPA Vacuum System Page – SVG Example



HIPA RF System Page – SVG + Plot Example



SwissFEL Camera Page – Web Component Example



Using Web Components enable us to create very clean html pages

But creating a web component control system library would require significant effort

Web-based approaches – balance sheet

Web Project - Benefits

- Web technologies offer the leading paradigm for today's software development. If we ignore them our controls solutions will be perceived as increasingly outdated and stuck in a timewarp ! Or worse, our users may simply choose to go elsewhere.
- Richer state-of-the-art UX possibilities are now easily achievable. For example: animations, sound, speech synthesis, speech recognition, messaging can be integrated very quickly.
- Since most of our web-based solutions are based on REST interfaces we get client side scripting possibilities for free (eg via curl).
- Web technologies fit the demands of modern society: remote working flexible/hybrid work models; access anywhere on multiple platforms, onsite, whilst travelling, or at home.
- Web technologies help our *Pikett* (on-call) support staff diagnose faults more quickly and efficiently, reducing down time.
- The pool of trained web developers is much larger than the pool of, say, EPICS or TANGO developers. Using web technologies should make our recruitment easier.

Web Projects - Challenges

- Richer, state-of-the art user experiences don't come without effort. Are our labs willing to pay that price, or do they prefer to invest the effort elsewhere ?
- Rich web applications tend to be created programmatically by software developers rather than by domain experts. Software engineers are not an easily scalable resource !
- Currently at PSI, we have no Web GUI builder tools to offer to our users. Such a tool would be necessarily complex (if we hope to deliver richer UX) or rely on simplifying abstractions (where we throw away some of the advantages of the web).
- Users may be reluctant to give up their old GUIs, even bad ones: "better the devil you know" ! The users sometimes resist our upgrade attempts unless there is a tangible payoff for them.
- We have no automatic conversion tool for our existing 5000+ caQtDm panels. Should we even create a tool which would only clone the user interface experiences of the past with their more limited richness ?
- The web is still in its infancy: as we try out and discard new approaches the rate of change is far higher than with our classical software development paradigms. There is an "impedance mismatch" between the lifecycle of web technologies and the lifecycle we would wish for the components used on our machines.

Web Projects – Future PSI Directions

The PSI Controls Group is currently very resource constrained and for the moment it cannot invest heavily in web development. At the moment we have no frontend team. ☹

Nevertheless, there are a number of web projects that are ongoing and we are hoping that our resource situation may change.

Current projects:

- Upgrade of the SwissFEL data acquisition and archiving systems
- Upgrade of the HIPA and PROSCAN interlock systems
- Continue to use WICA to bring strategically chosen control system displays to the web in support of our Pikett teams and other remote workers

I am still pursuing the dream of a tool that would enable our users to create their own displays for the web ! ☺

Thank you for your attention :-)

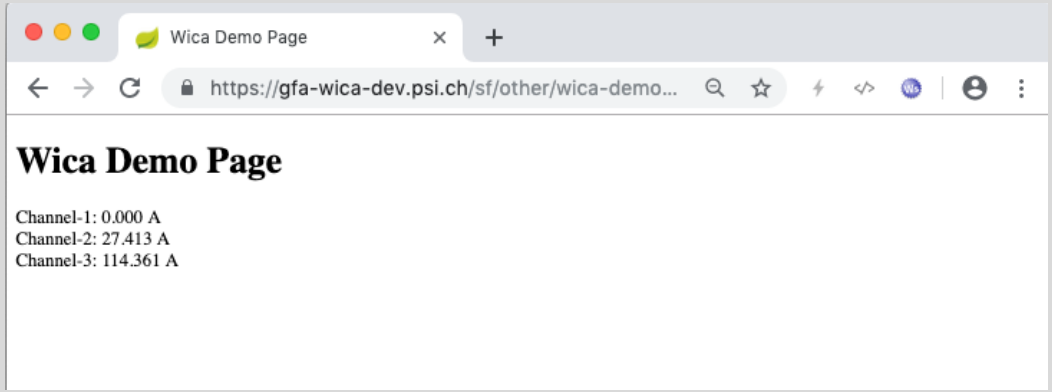
A particular thank you to Chris Roderick for setting up this workshop

and to

Simon Ebner for making suggestions to these slides

Simple Wica Web Page Example

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8"/>
    <title>Wica Demo Page</title>
    <script src="/wica/wica.js" type="module"></script>
</head>
<body>
    <h1>Wica Demo Page</h1>
    <label>Channel-1:</label> <span data-wica-channel-name="SINEG01-MBND300:I-READ"></span> <br>
    <label>Channel-2:</label> <span data-wica-channel-name="SINLH02-MBND100:I-READ"></span> <br>
    <label>Channel-3:</label> <span data-wica-channel-name="SINBC02-MBND100:I-READ"></span> <br>
</body>
</html>
```



Attempting the risky – a live demo of WICA ?

Projects

Our projects vary enormously in size, scope and complexity…

Green Field Projects

Example

“Build a Free Electron Laser”

- Freedom to use new and improved technologies ☺
- Often a top priority of the institute => more resources ☺
- Plenty of engagement from the users and scientists ☺
- Requirements less clear – we start off not knowing where the design journey will end
- Time pressure can lead to settling for existing safe solutions rather than moving onwards to new technologies ☹

Projects

Our projects vary enormously in size, scope and complexity…

A never-ending task: painting the Forth Railway Bridge

Upgrade Projects

Example

"Upgrade the Machine Safety System at HIPA"

- Characteristics of the design/problem space are typically better understood ☺
- Reduced time pressure (as long as the old solution still works) ☺
- Less engagement from the users (unless there are significant feature enhancements) ☹
- Less resources to do the work ☹
- May need to migrate in steps => things can get more complicated in short term ☹

Footnote: in 2009 the use of new paint technology meant that it finally became possible to make breaks in the painting cycles !