PAUL SCHERRER INSTITUT

PSI

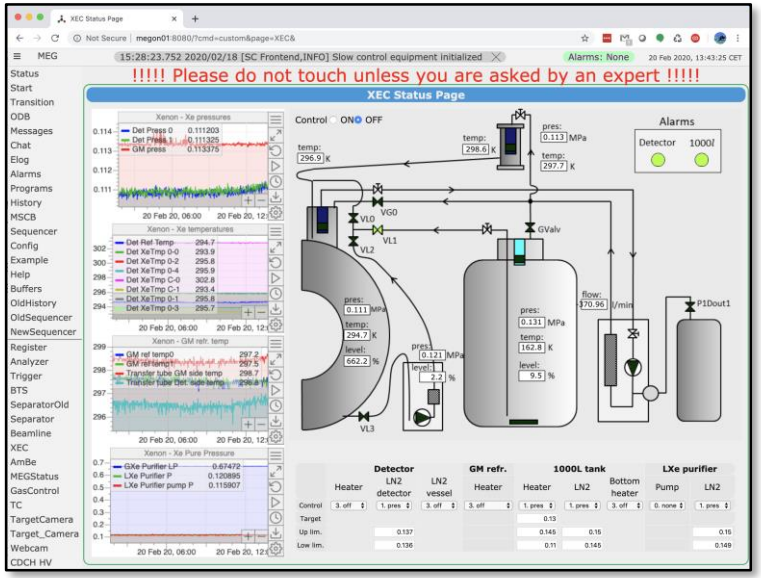
Stefan Ritt :: Head Muon Physics :: Paul Scherrer Institute

Real-time Web Display in the MIDAS DAQ Software

22nd IEEE Virtual Real Time Conference

Motivation

- Experiments need graphical user interface
- Native applications are hard to handle
  - Need to compile separately for different architectures
  - Need to distribute new versions
  - Not available on smart devices
- Wouldn't it be nice to run everything from your web browser?
  - How to handle graphics?
  - Real-time updates?

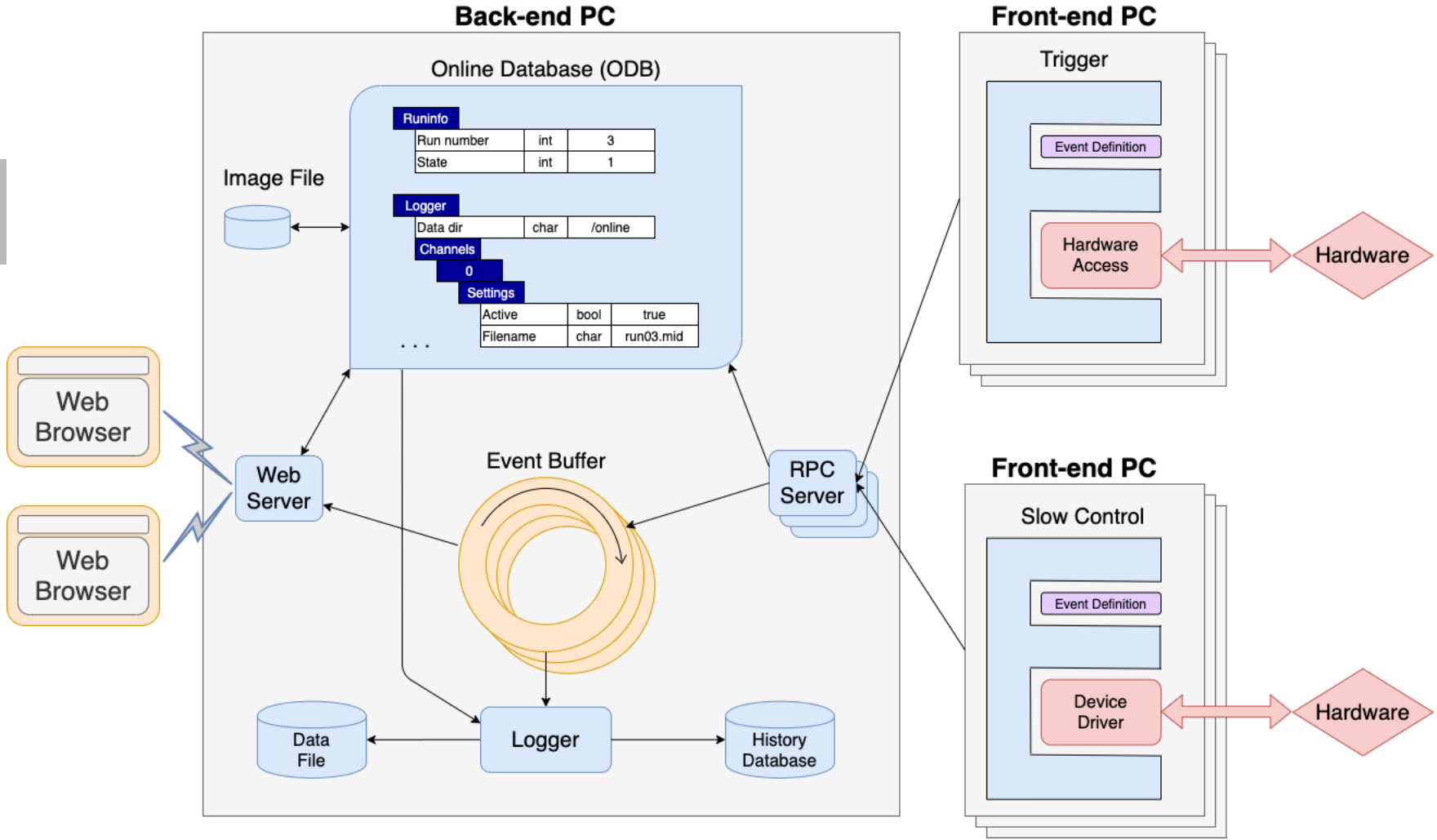


MIDAS DAQ System

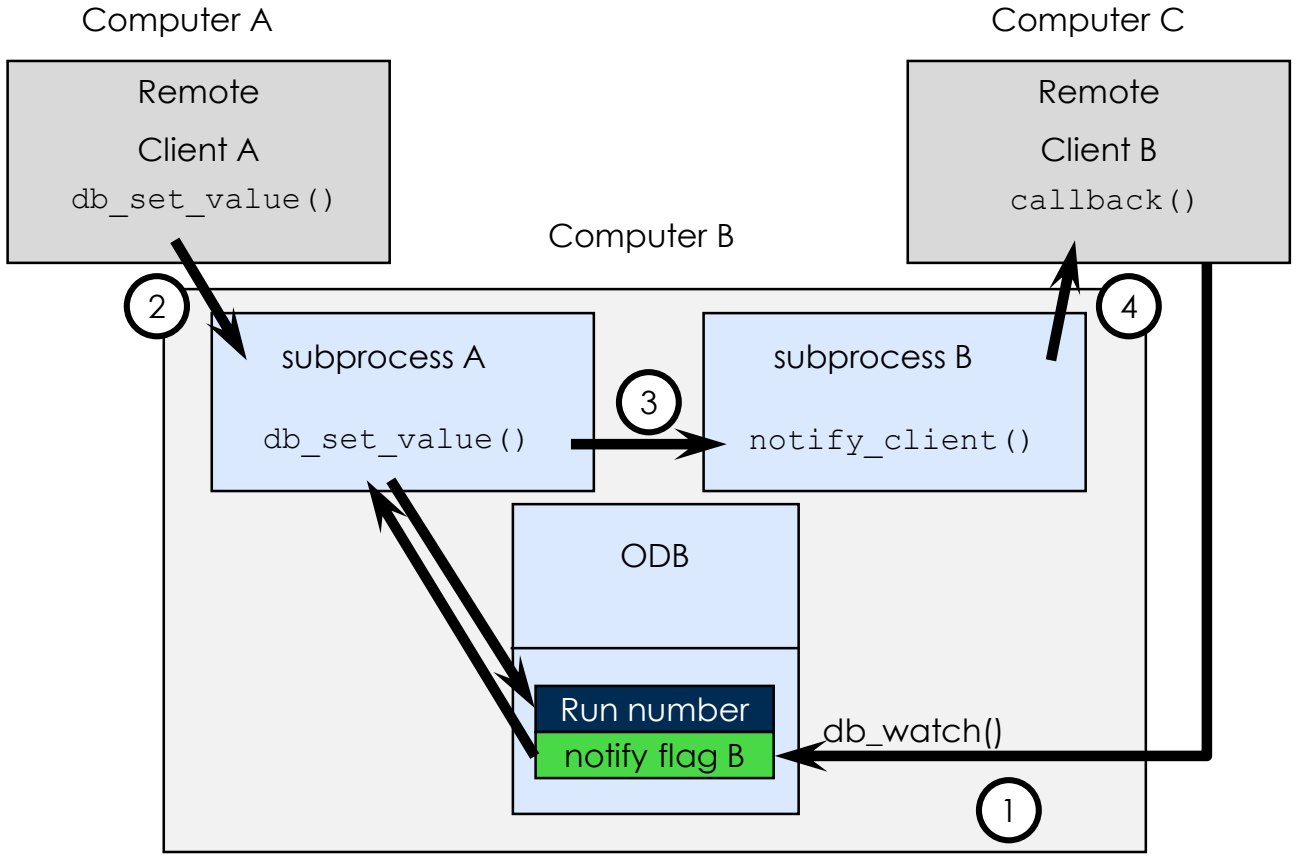
- Fulfill needs of **small** and **medium** size **experiments**
  (~1000's of channels, 100's MB data/sec.)
  - Incorporated **slow control**
    (Maximum Integrated Data Acquisition System)
  - Integrated **data analysis** ("histogramming")
  - Operating system / hardware **independent**
  - ~150'000 lines of C / C++ / JavaScript code
  - Quick installation
  - Easy application programming
  - Free (GLP)

- Development started in **1993** at PSI, CH and 1996 at TRIUMF, CA

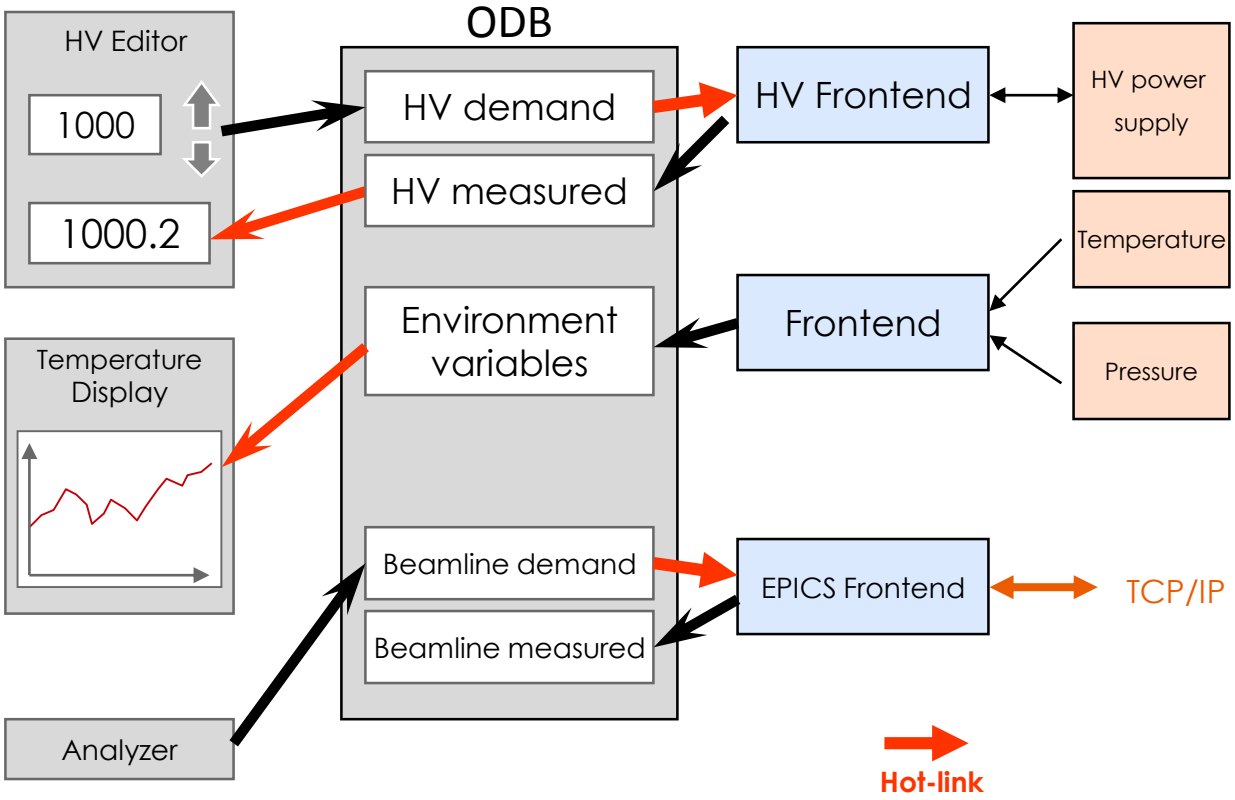
- Today used at **CERN** (alpha), **KEK** (T2K), **Fermilab** (g-2),
  TRIUMF, **PSI** (standard DAQ system in particle physics)


Back-end PC
Online Database (ODB)
Runinfo
Run number int 3
State int 1
Logger
Data dir char /online
Channels
0
Settings
Active bool true
Filename char run03.mid
. . .
Image File
Web Browser
Web Browser
Web Server
Event Buffer
RPC Server
Data File
Logger
History Database
Front-end PC
Trigger
Event Definition
Hardware Access
Hardware
Front-end PC
Slow Control
Event Definition
Device Driver
Hardware

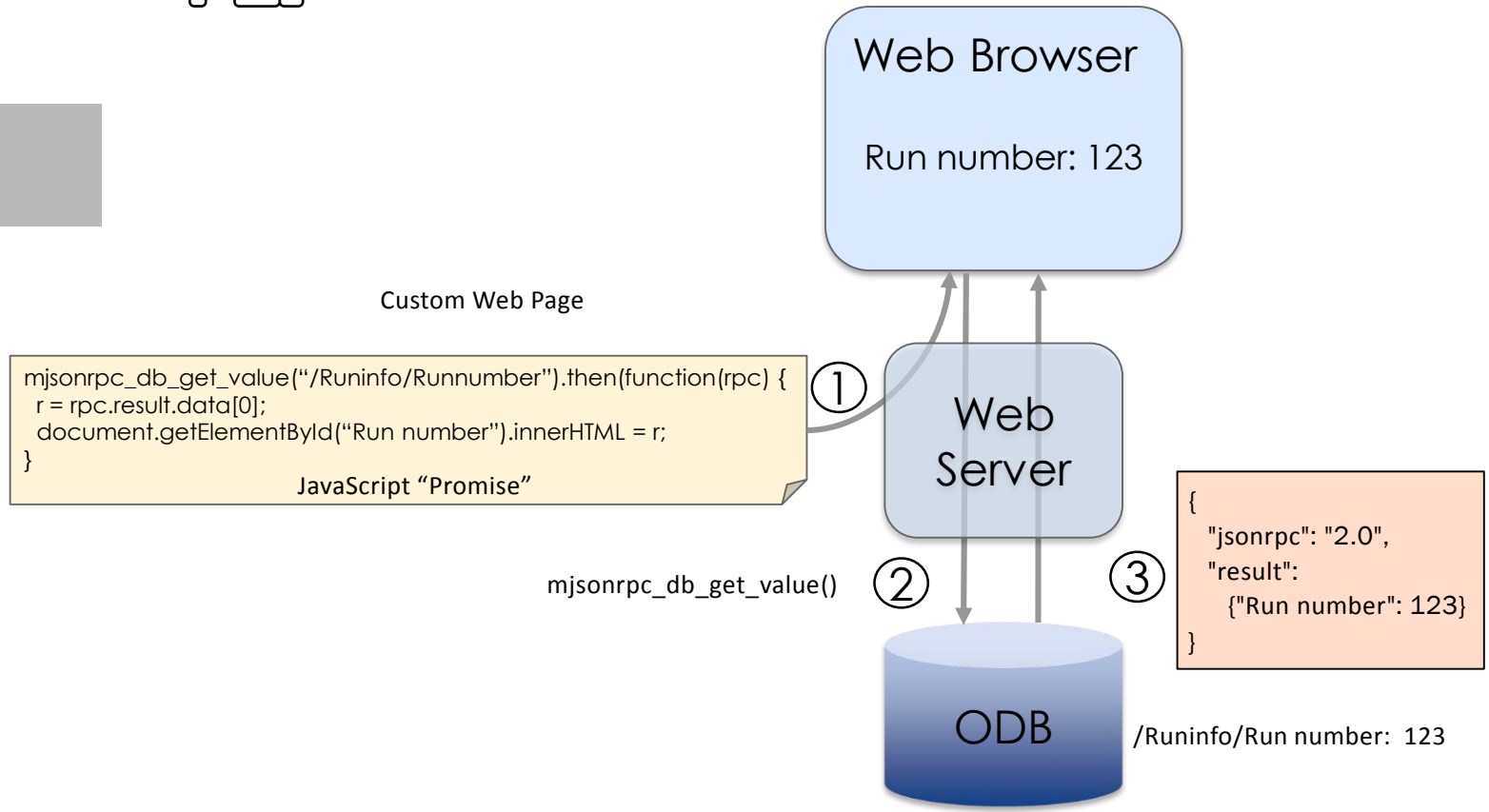
Data change notification ("**hot link**")



Slow Control Through ODB Hot-links



MJSON-RPC


Web Browser
Run number: 123
Custom Web Page
mjsonrpc_db_get_value("/Runinfo/Runnumber").then(function(rpc) {
r = rpc.result.data[0];
document.getElementById("Run number").innerHTML = r;
}
JavaScript "Promise"
1
Web Server
mjsonrpc_db_get_value()
2
3
{
"jsonrpc": "2.0",
"result":
{"Run number": 123}
}
ODB
/Runinfo/Run number: 123

How to do Web programming

- Use new HTML5 features: AJAX, JSON, Canvas
  - JavaScript is today as fast as a native application a few years ago

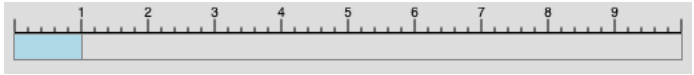
- Do NOT rely on 3rd party libraries (Angular, React, …)
  - Minimize number of dependencies
  - Minimize and optimize network transfer
  - Minimize user learning
  - Be safe for the future!

- Use standard HTML and extend it
  <div data-xxxx /> → access in JS via element.dataset.xxxx

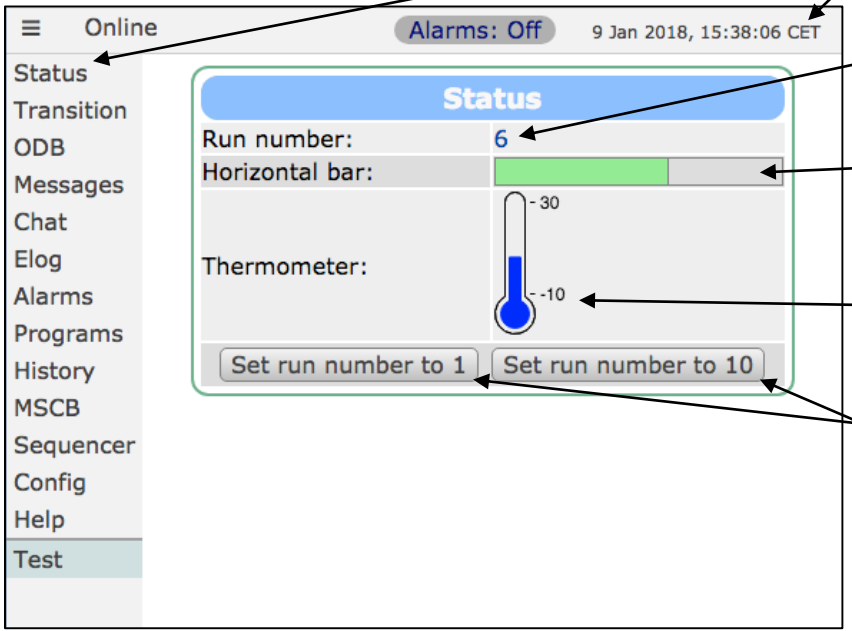
Green: standard HTML

Blue: MIDAS extensions

```
<div class="modbhbar" style="width: 200px; height: 18px;"
   data-odb-path="/Runinfo/Run number"
   data-max-value="10"></div>
```



Complete example

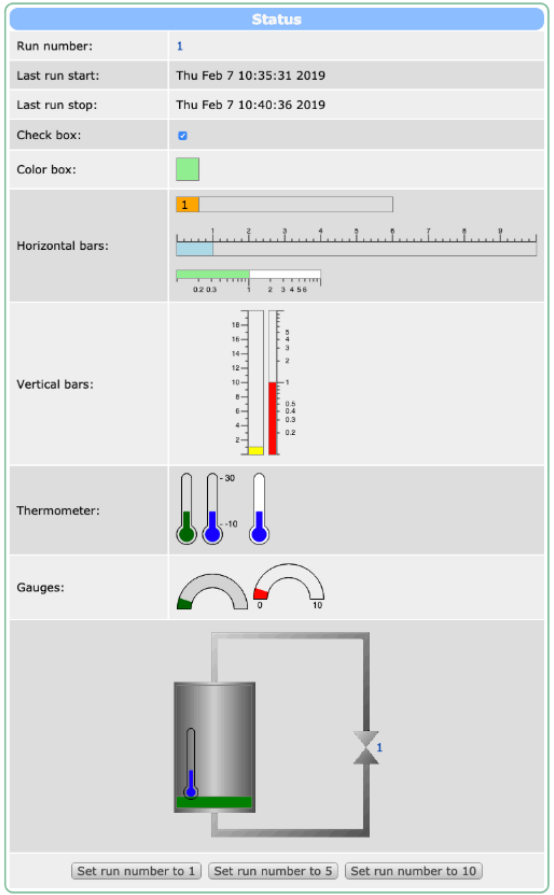



```
<html class="mcss">
<head>
<link rel="stylesheet" href="midas.css">
  <script src="controls.js"></script>
  <script src="midas.js"></script>
  <script src="mhttpd.js"></script>
  <title>Example</title>
</head>

<body class="mcss" onload="mhttpd_init('Test', 1000);">
<div id="mheader"></div>
<div id="msidenav"></div>
<div id="mmain">
  <table class="mtable">
    <tr><th colspan="2" class="mtableheader">Status</th></tr>
    <tr>
      <td style="width: 200px;">Run number:</td>
      <td>
        <div class="modbvalue" data-odb-path="/Runinfo/Run number"
         data-odb-editable="1"></div>
      </td>
    </tr>
    <tr>
      <td>Horizontal bar:</td>
      <td>
        <div class="modbhbar" style="width: 200px; height: 18px;"
         data-odb-path="/Runinfo/Run number" data-max-value="10"
         data-color="lightgreen"></div>
      </td>
    </tr>
    <tr>
      <td>Thermometer:</td>
      <td>
        <div class="modbthermo" style="width:60px;height:100px;"
         data-odb-path="/Runinfo/Run number" data-min-value="-10"
         data-max-value="30" data-color="blue" data-scale="1"></div>
      </td>
    </tr>
    <tr>
      <td colspan="2" style="text-align: center;">
        <button class="modbbutton" class="mbutton"
         data-odb-path="/Runinfo/Run number" data-odb-value="1">
         Set run number to 1</button>
        <button class="modbbutton" class="mbutton"
         data-odb-path="/Runinfo/Run number" data-odb-value="10">
         Set run number to 10</button>
      </td>
    </tr>
  </table>
</div>
</body>
</html>
```

MIDAS code and styles

https://midas.triumf.ca/MidasWiki/index.php/Custom_Page



modbvalue

modbcheckbox
modbbox

modbhbar + mhaxis

modbvbar + mvaxis

modbthermo

modbgauge

modbthermo + modbvbar
on top of image

modbbutton

Pooling of AJAX requests

Request:

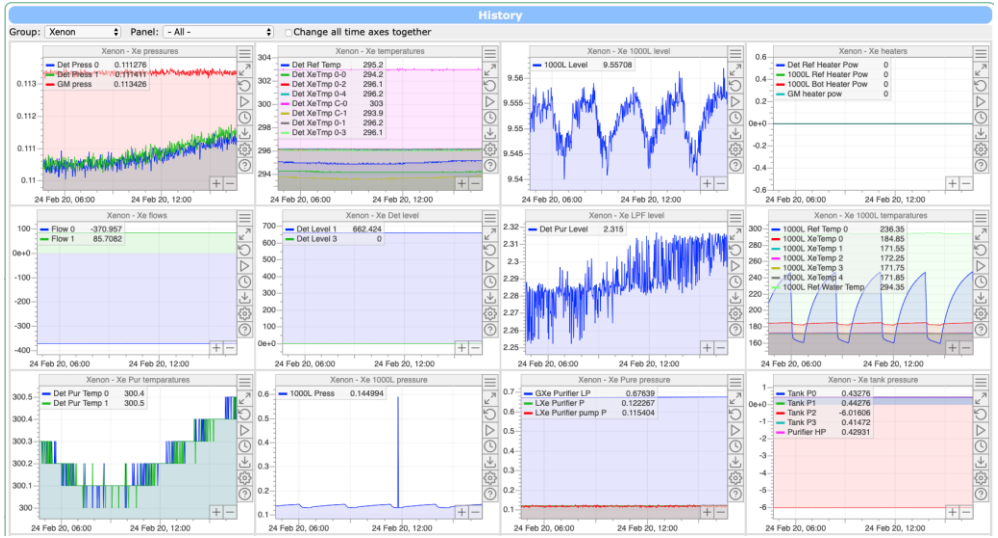
```
{
  "jsonrpc":"2.0",
  "method":"db_get_values",
  "params": {
    "paths":[
  ①  "/Logger/Write data",
  ②  "/Runinfo/Run number",
  ③  "/Runinfo/Start time",
      ]
  },
  "id":1582556170492
}
```

Reply:

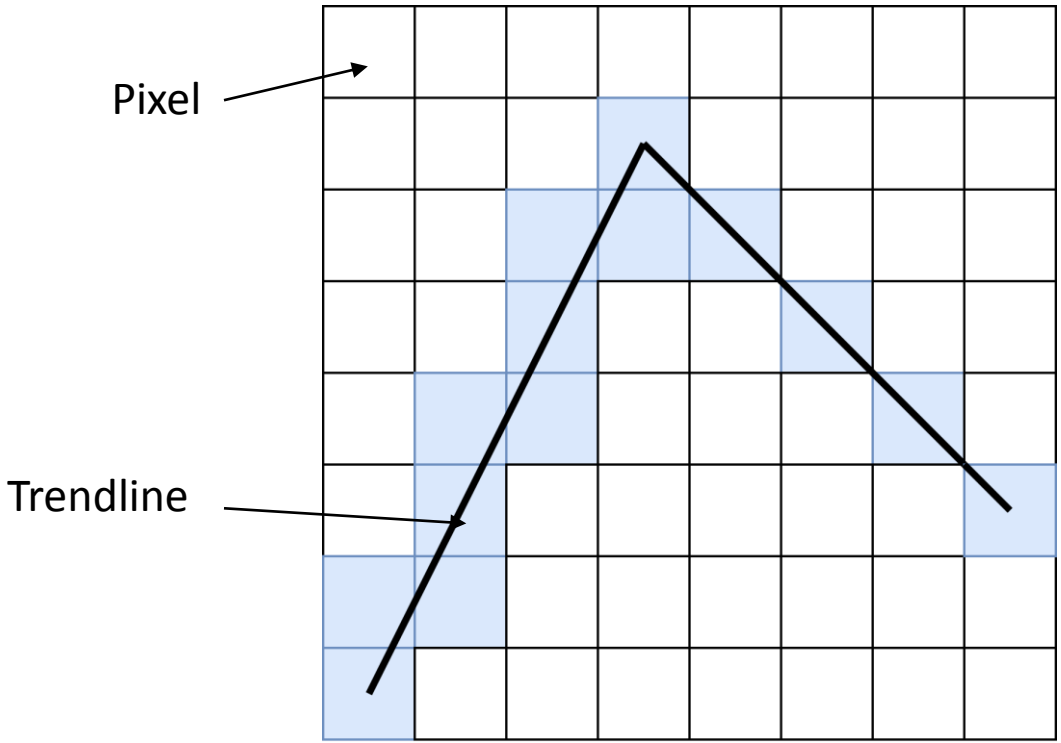
```
{
  "result": {
    "data":[
  ① false,
  ② 123,
  ③ "Fri Feb  7 15:25:12 2020"
    ]
  },
  "jsonrpc":"2.0",
  "id":1.582556170492
}
```

History Panel

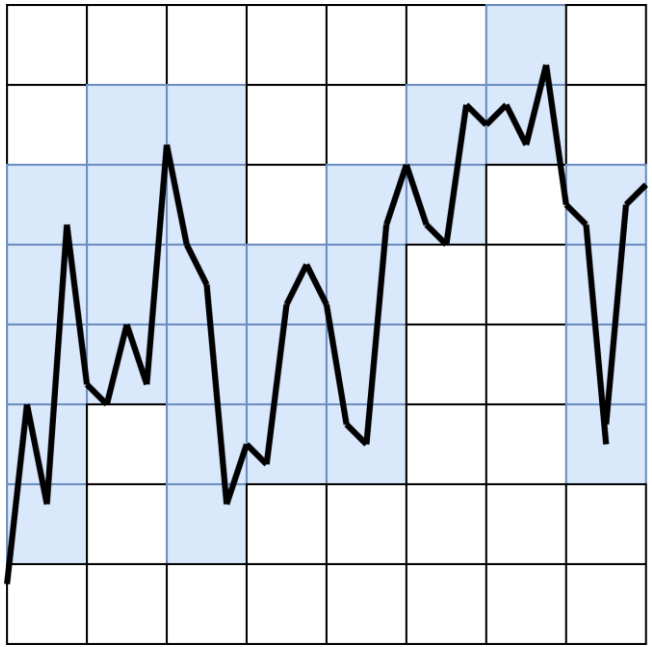
- GUI to render "history" of experiment variables
- Written in pure JavaScript using HTML5 canvas
- Intuitive interface for scrolling and zooming
- Switching to JavaScript "typed arrays" 5x faster
- Optimization to render millions of points

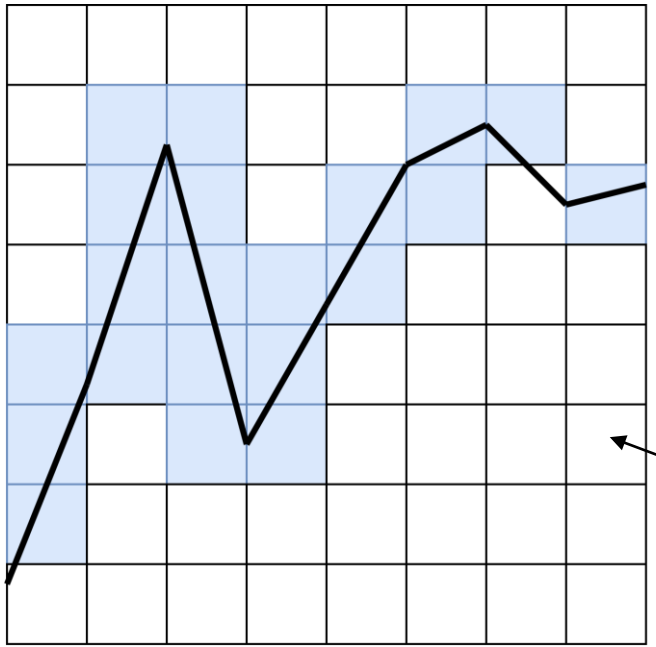


How to draw a trend-line

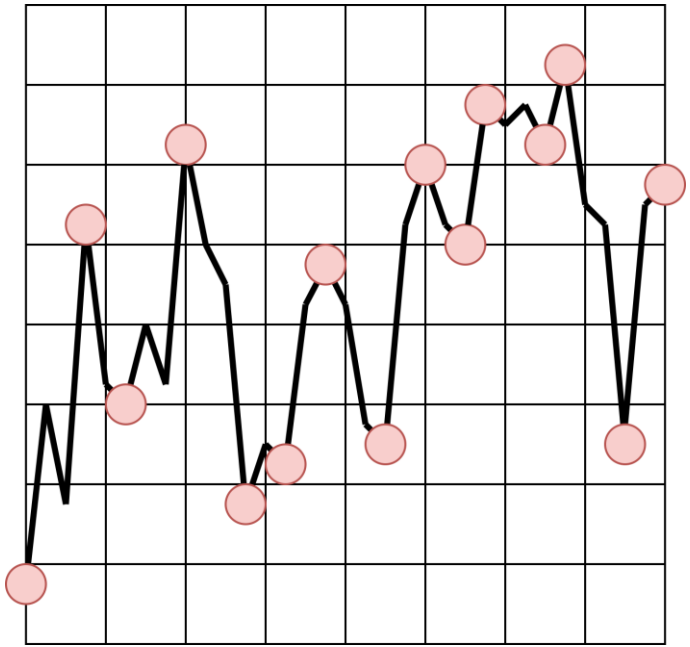


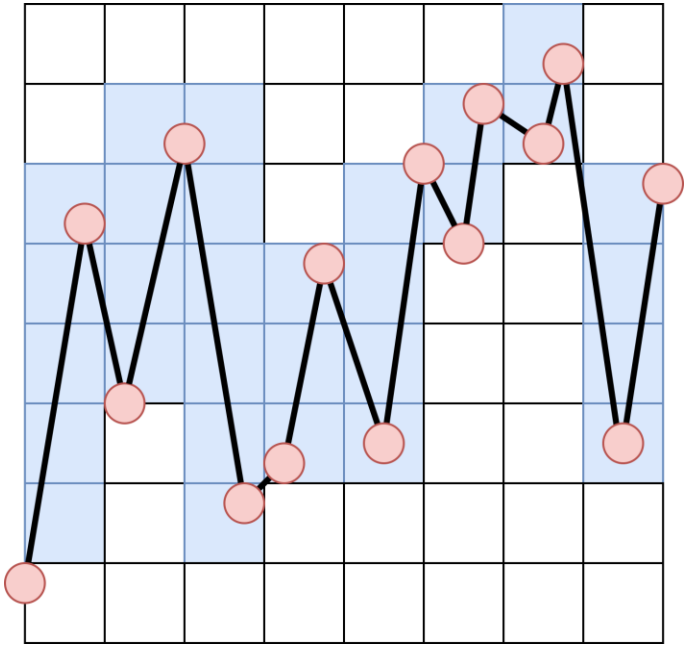
Decimation can conceal important data



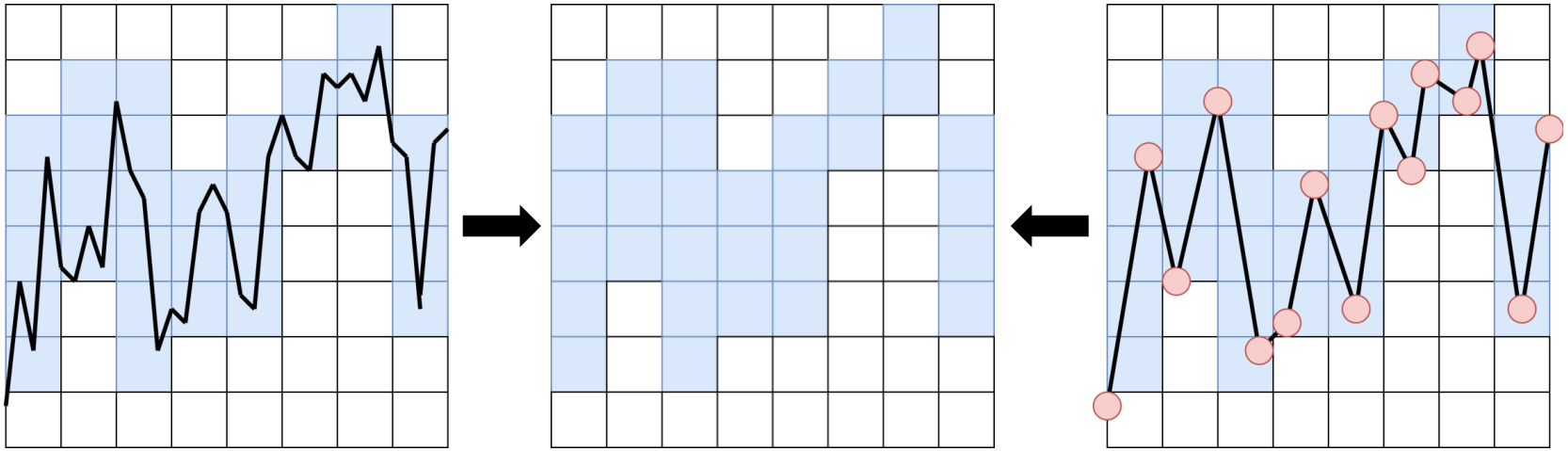


Solution: Search for min & max in each pixel column and connect with lines





Min-max drawing does not conceal data



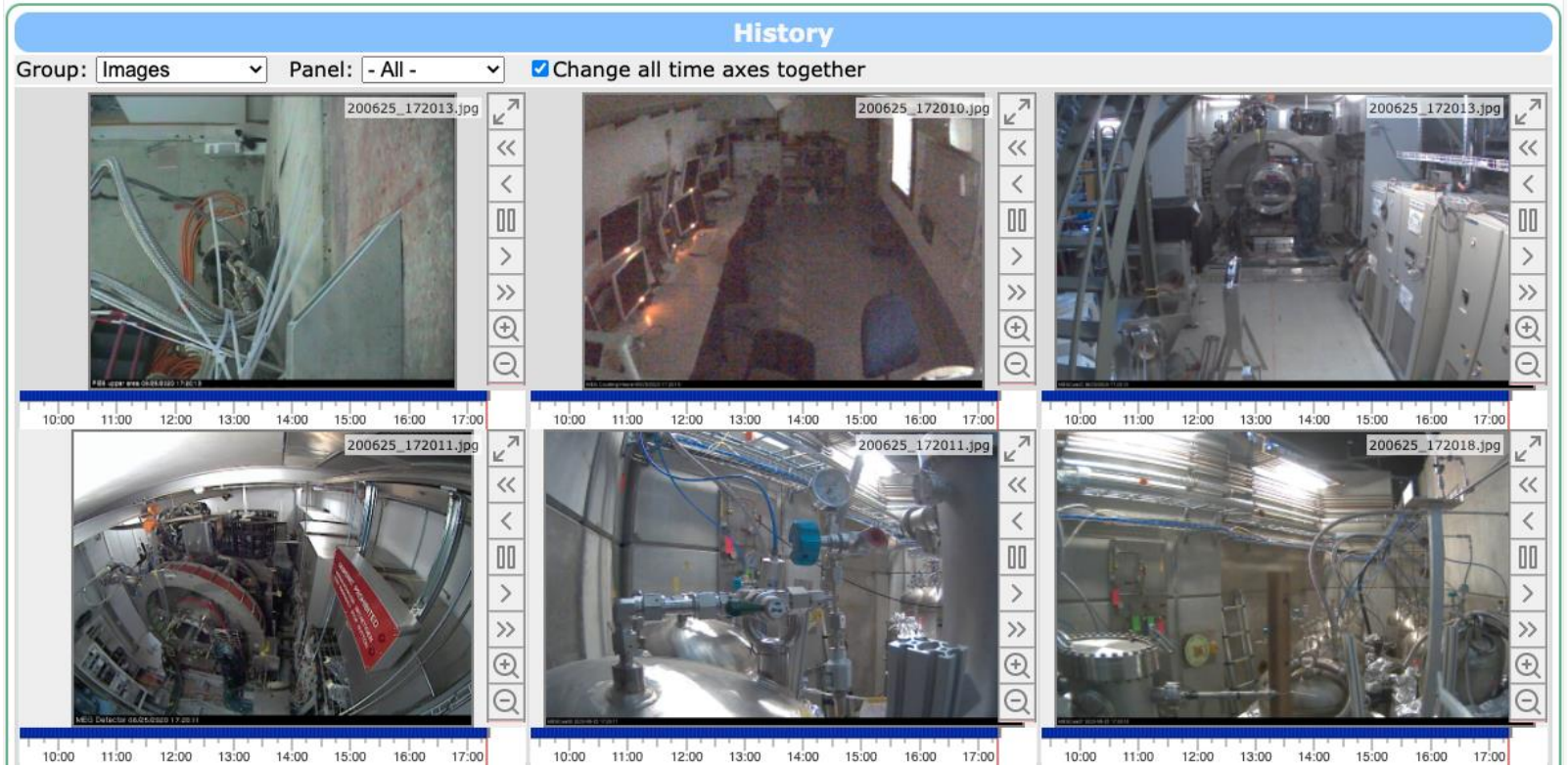
Comparison of drawing methods

1 year data, 3 values, one sample per minute → 1.5 M points

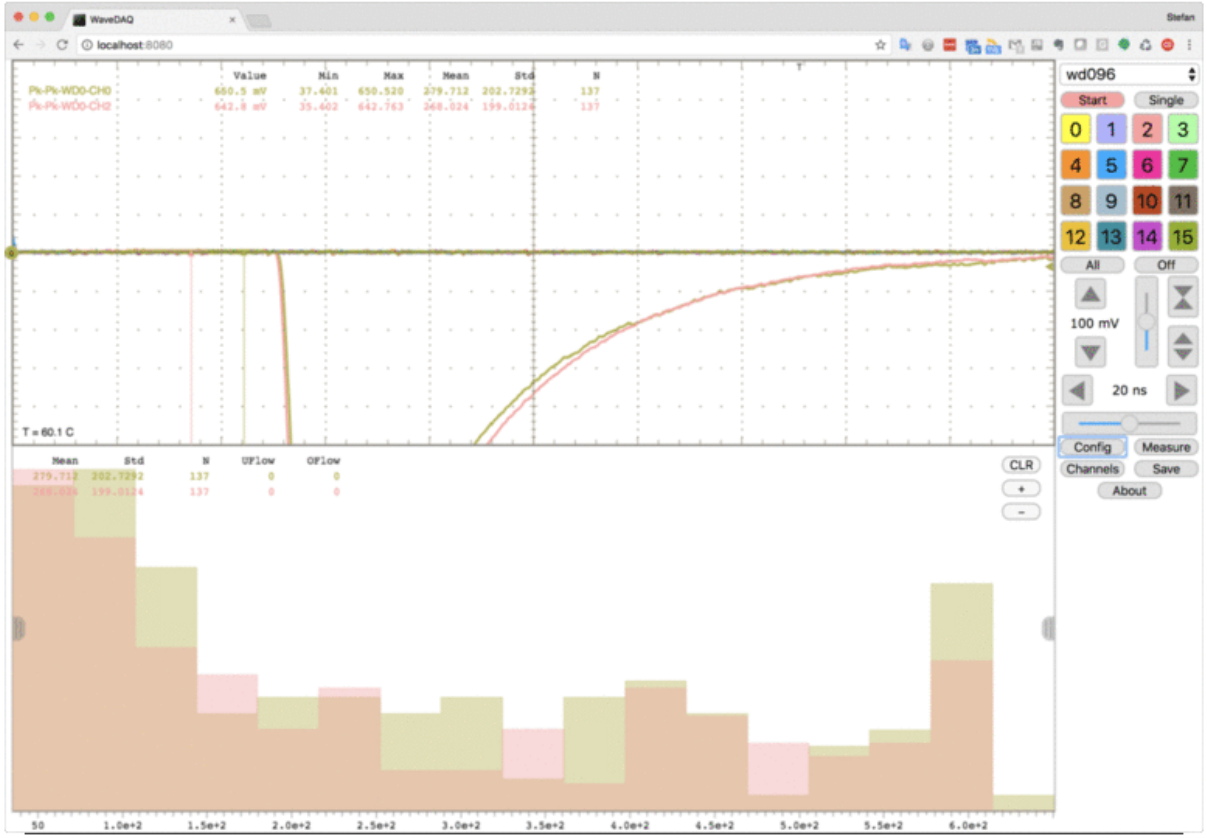
Normal method: 90 ms

New method: 3 ms

Image History

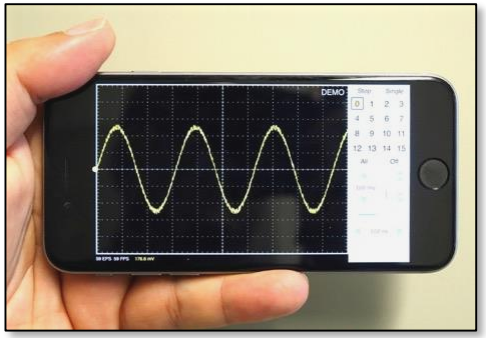


Waveform display in browser with HTML5



http://elog.psi.ch/scope

- 70 frames/s display rate
- Integrated measurements with histogramming
- Run on any browser
- Planned to integrate into MIDAS



Conclusions

- Many thanks to Konstantin Olchanski (TRIUMF)
- MIDAS: http://midas.triumf.ca
- Slow control part of MIDAS can be used stand-alone