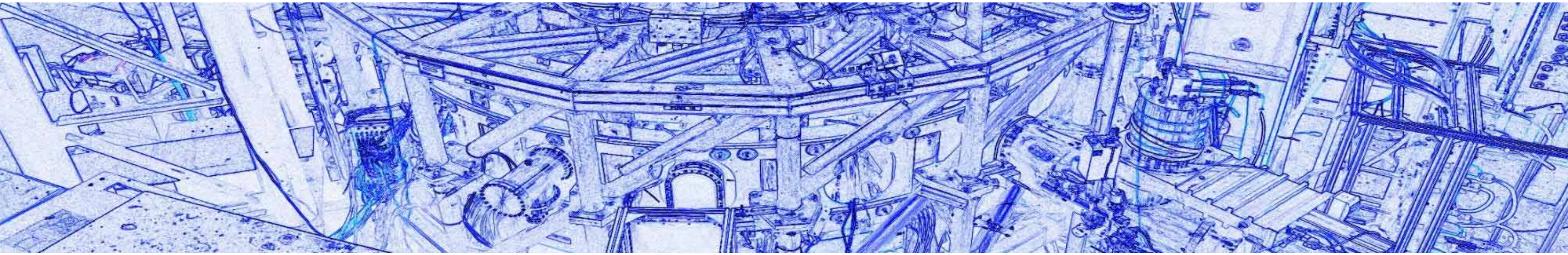
磁约束聚变与等离子体国际合作联合实验室
International Joint Research Laboratory of Magnetic Confinement Fusion and Plasma Physics

Web Technology Enabling Fast and Easy Large Experiment Facility Control System Implementation

Zheng Wei and J-TEXT Team

ICALEPCS 2023



Outline

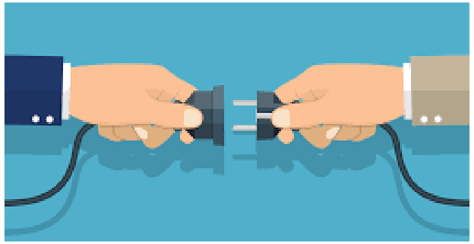
- **Introduction on control system based on web technology**
- **A software toolkit for building a control system using web technology**
- **Leverage on existing technology enabling fast and easy control system implementation**
- **Real world applications**

Outline

- **Introduction on control system based on web technology**
- A software toolkit for building a control system using web technology
- Leverage on existing technology enabling fast and easy control system implementation
- Real world applications

Why we may need web technology?

Interoperability



- Everything needs to understand and work with others
- Everything needs be integrated effortlessly
- We need them to speak a common language

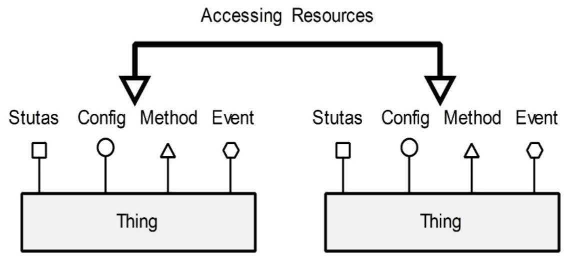
The Web Technology

- **What is web technology and why?**
  - HTTP
    - The common language for communication
  - HTML
    - The common language for visualization
  - Browsers
    - The common language for user interaction

Web is designed for interoperability

Control (SCADA) System Abstraction

- **Resources**
  - Thing： Something that contains other resources
  - Status: Information that a thing wants to expose—read only
  - Configuration: Indicate the desired behavior by other—read & write
  - Method: A command—immediate action
  - Event: When subscribed, will invoke a method on certain condition
- **Access of resources—RESTful Web API**
  - Using URL to identify the resources
  - Using HTTP verb to specify the action
  - HTTP response as the result sample



Resource Access Action	HTTP Verb
Get	GET
Set	Put
Invoke	Post
Subscribe	Post
Unsubscribe	Delete

Control system based on Web technology

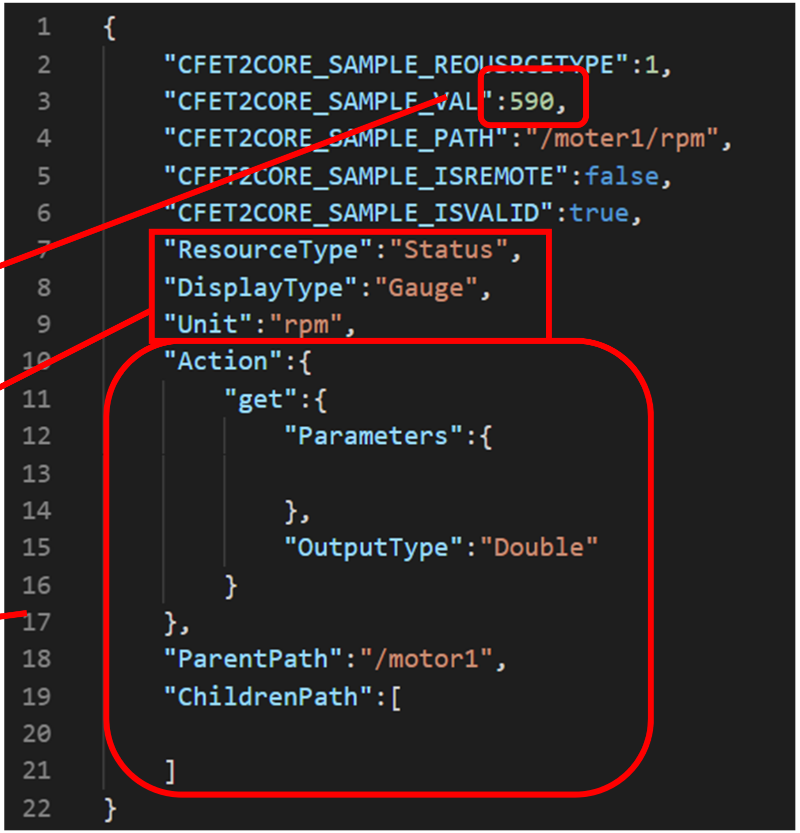
- **A request to a control system resource**

- **Get: http://pulsegenerator.powersystem.local/motor1/rpm**

The actual Value of this status

Hypermedia Data for Visualization

Hypermedia data for Navigation
and how to interact with this resource



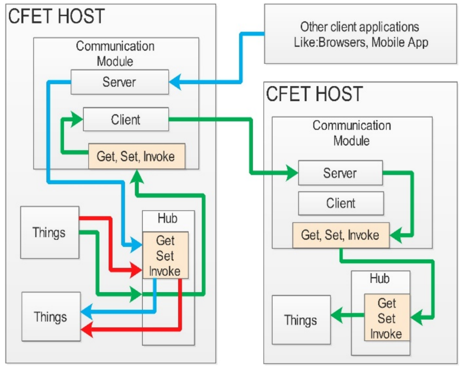
```
{
    "CFET2CORE_SAMPLE_REQUSRCETYPE":1,
    "CFET2CORE_SAMPLE_VAL":590,
    "CFET2CORE_SAMPLE_PATH":"/moter1/rpm",
    "CFET2CORE_SAMPLE_ISREMOTE":false,
    "CFET2CORE_SAMPLE_ISVALID":true,
    "ResourceType":"Status",
    "DisplayType":"Gauge",
    "Unit":"rpm",
    "Action":{
        "get":{
            "Parameters":{

            },
            "OutputType":"Double"
        }
    },
    "ParentPath":"/motor1",
    "ChildrenPath":[

    ]
}
```

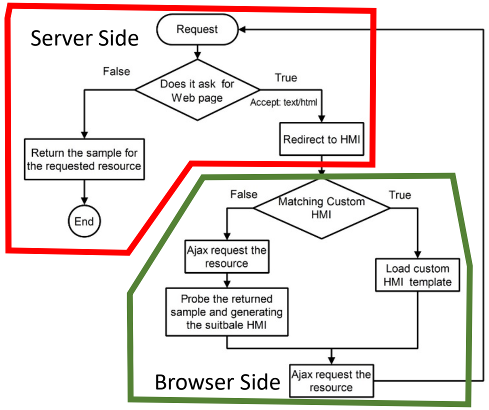
Outline

- Introduction on control system based on web technology
- **A software toolkit for building a control system using web technology**
- Leverage on existing technology enabling fast and easy control system implementation
- Real world applications

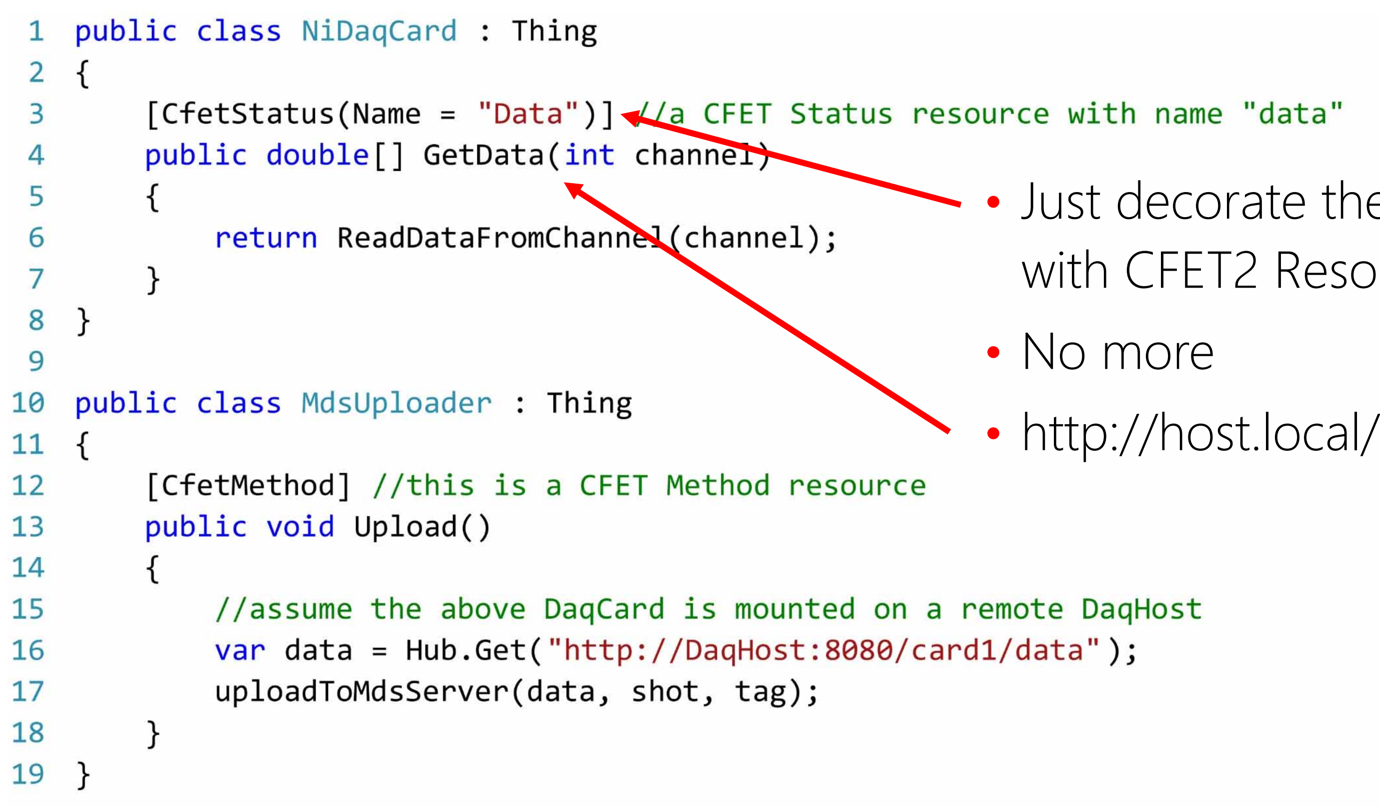
The CFET Toolkit

- **OK, you can build a control system with web technology using all kinds of web server and client libraries.**
- **But we created a software toolkit for that.**
- **CFET2**
  - A console app called CFET2app, just like an EPICS IOC
  - A standalone Single Page Application for HMI design called WidgetUI




Server Side
Request
False
Does it ask for Web page
True
Accept: text/html
Return the sample for the requested resource
Redirect to HMI
End
False
Matching Custom HMI
True
Ajax request the resource
Load custom HMI template
Probe the returned sample and generating the suitbale HMI
Browser Side
Ajax request the resource

How to make a Thing?



```
public class NiDaqCard : Thing
{
    [CfetStatus(Name = "Data")] //a CFET Status resource with name "data"
    public double[] GetData(int channel)
    {
        return ReadDataFromChannel(channel);
    }
}

public class MdsUploader : Thing
{
    [CfetMethod] //this is a CFET Method resource
    public void Upload()
    {
        //assume the above DaqCard is mounted on a remote DaqHost
        var data = Hub.Get("http://DaqHost:8080/card1/data");
        uploadToMdsServer(data, shot, tag);
    }
}
```

- Just decorate the methods/properties with CFET2 Resource Attribute
- No more
- http://host.local/daq/card1/data/5

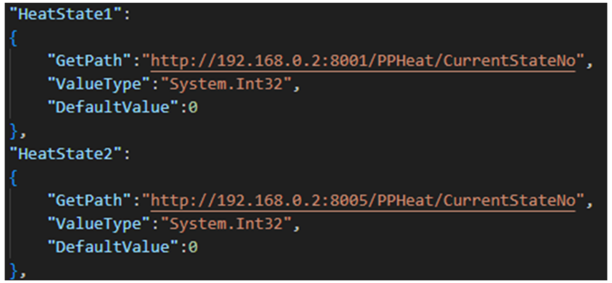
How to make an HMI

- **By drag and drop**
  - Wid
  - you
  - You

[redacted]

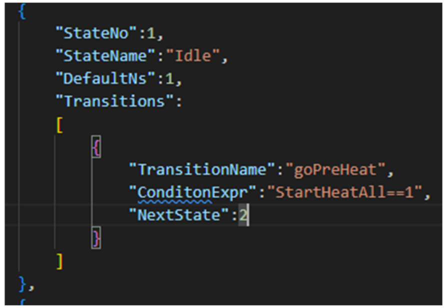
Some Important Things

- **State machine thing**
  - Config with 3 files： Aliases, State transitions, Actions



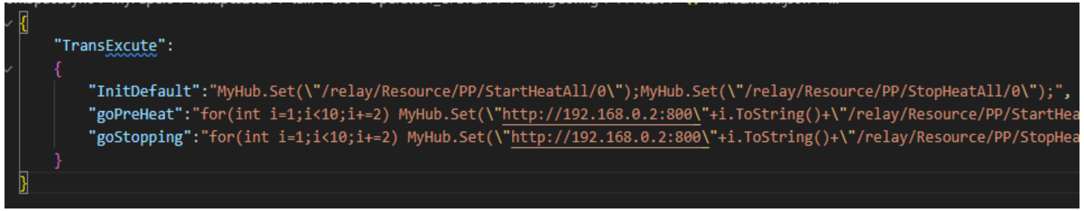
```
"HeatState1":
{
    "GetPath":"http://192.168.0.2:8001/PPHeat/CurrentStateNo",
    "ValueType":"System.Int32",
    "DefaultValue":0
},
"HeatState2":
{
    "GetPath":"http://192.168.0.2:8005/PPHeat/CurrentStateNo",
    "ValueType":"System.Int32",
    "DefaultValue":0
},
```

Aliases： URL to variable name mapping



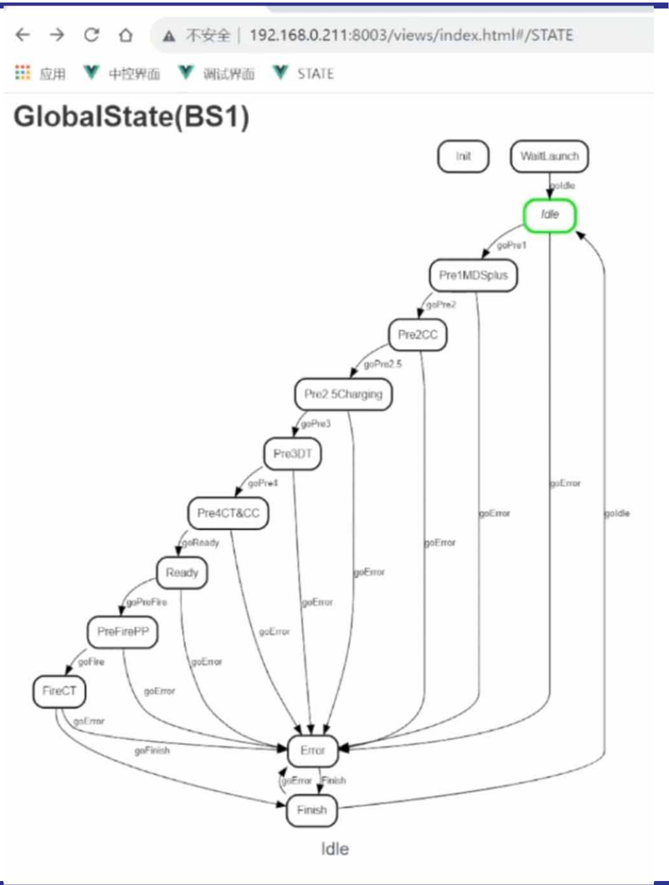
```
{
    "StateNo":1,
    "StateName":"Idle",
    "DefaultNs":1,
    "Transitions":
    [
        {
            "TransitionName":"goPreHeat",
            "ConditonExpr":"StartHeatAll==1",
            "NextState":2
        }
    ]
},
```

State transitions： defines states and transition conditions



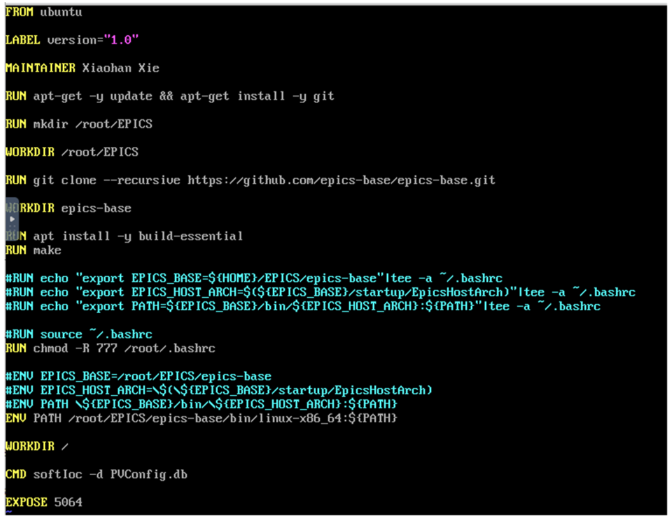
```
{
    "TransExcute":
    {
        "InitDefault":"MyHub.Set(\"/relay/Resource/PP/StartHeatAll/0\");MyHub.Set(\"/relay/Resource/PP/StopHeatAll/0\");",
        "goPreHeat":"for(int i=1;i<10;i+=2) MyHub.Set(\"http://192.168.0.2:800\"+i.ToString()+\"/relay/Resource/PP/StartHea
        "goStopping":"for(int i=1;i<10;i+=2) MyHub.Set(\"http://192.168.0.2:800\"+i.ToString()+\"/relay/Resource/PP/StopHea
    }
}
```

Actions： what to do when a transition fires

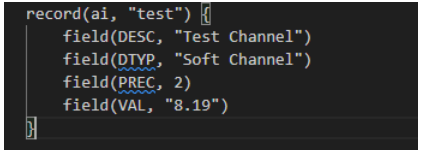


EPICS Bridget thing

- **Associate a CFET2 resource (a RESTful api) with an EPICS PV**
  - Allow EPICS clients to access CFET2 resource
  - Allow HTTP client to access EPICS PVs
  - Made a soft IOC docker image to make everything simple



```
FROM ubuntu

LABEL version="1.0"

MAINTAINER Xiaohan Xie

RUN apt-get -y update && apt-get install -y git

RUN mkdir /root/EPICS

WORKDIR /root/EPICS

RUN git clone --recursive https://github.com/epics-base/epics-base.git

WORKDIR epics-base

RUN apt install -y build-essential
RUN make

#RUN echo "export EPICS_BASE=${HOME}/EPICS/epics-base"|tee -a ~/.bashrc
#RUN echo "export EPICS_HOST_ARCH=$(${EPICS_BASE}/startup/EpicsHostArch)"|tee -a ~/.bashrc
#RUN echo "export PATH=${EPICS_BASE}/bin/${EPICS_HOST_ARCH}:${PATH}"|tee -a ~/.bashrc

#RUN source ~/.bashrc
RUN chmod -R 777 /root/.bashrc

#ENV EPICS_BASE=/root/EPICS/epics-base
#ENV EPICS_HOST_ARCH=\$(\${EPICS_BASE}/startup/EpicsHostArch)
#ENV PATH \${EPICS_BASE}/bin/\${EPICS_HOST_ARCH}:${PATH}
ENV PATH /root/EPICS/epics-base/bin/linux-x86_64:${PATH}

WORKDIR /

CMD softIoc -d PVConfig.db

EXPOSE 5064
```



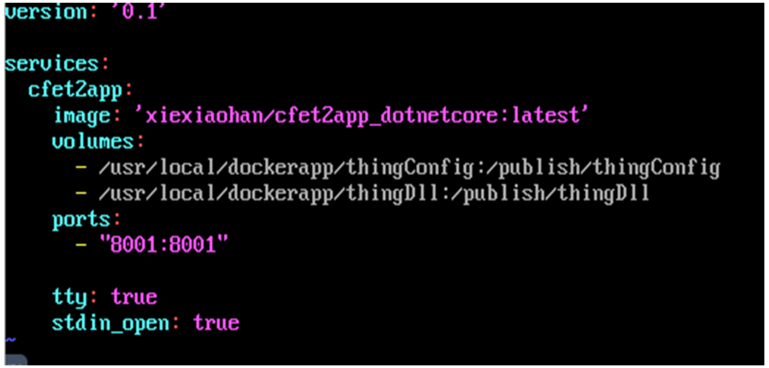
```
record(ai, "test") {
    field(DESC, "Test Channel")
    field(DTYP, "Soft Channel")
    field(PREC, 2)
    field(VAL, "8.19")
}
```

Outline

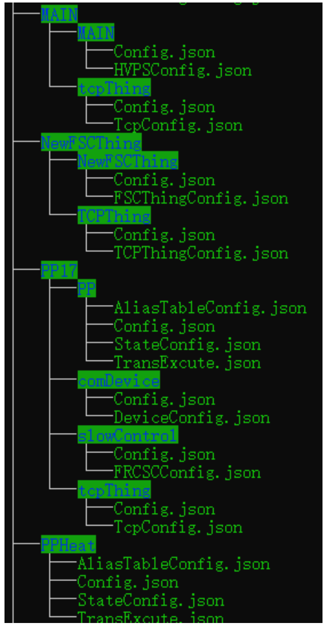
- Introduction on control system based on web technology
- A software toolkit for building a control system using web technology
- **Leverage on existing technology enabling fast and easy control system implementation**
- Real world applications

Docker

- **Docker enables fast deployment**
  - CFET2app can be regarded as Micro-services
  - Docker allow them to be deployed with minimum effort
    - Create folders fot Things
    - Put thing config files in the folder
    - Run the image



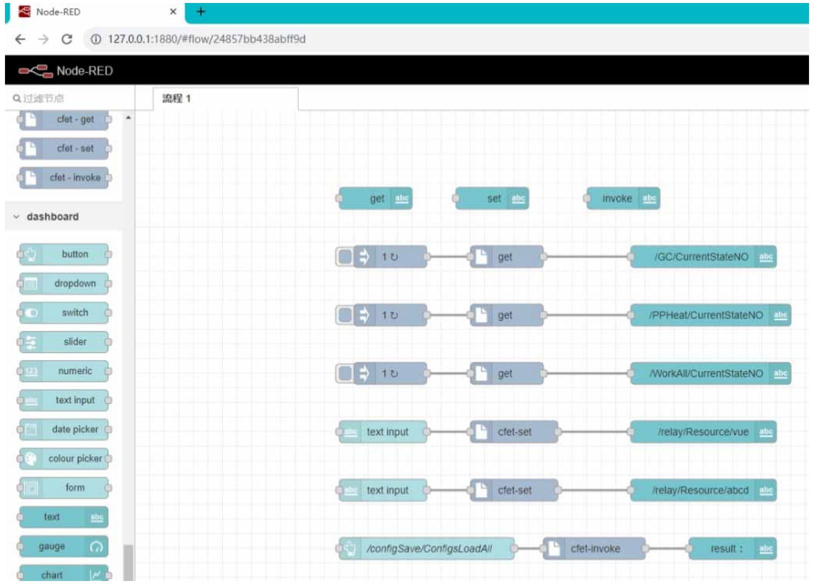
```
version: '0.1'

services:
  cfet2app:
    image: 'xiexiaohan/cfet2app_dotnetcore:latest'
    volumes:
      - /usr/local/dockerapp/thingConfig:/publish/thingConfig
      - /usr/local/dockerapp/thingDll:/publish/thingDll
    ports:
      - "8001:8001"

    tty: true
    stdin_open: true
~
```

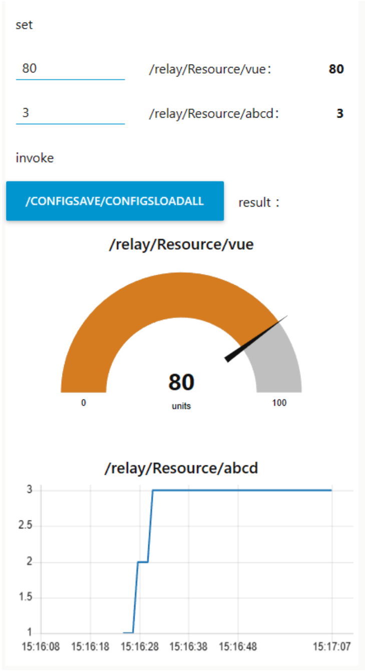


```
├─MAIN
│  ├─MAIN
│  │  ├─Config.json
│  │  └─HVPSConfig.json
│  └─tcpThing
│      ├─Config.json
│      └─TcpConfig.json
├─NewFSCThing
│  ├─NewFSCThing
│  │  ├─Config.json
│  │  └─FSCThingConfig.json
│  └─TCPThing
│      ├─Config.json
│      └─TCPThingConfig.json
├─PP17
│  ├─PP
│  │  ├─AliasTableConfig.json
│  │  ├─Config.json
│  │  ├─StateConfig.json
│  │  └─TransExcute.json
│  ├─comDevice
│  │  ├─Config.json
│  │  └─DeviceConfig.json
│  ├─slowControl
│  │  ├─Config.json
│  │  └─FRCSCConfig.json
│  └─tcpThing
│      ├─Config.json
│      └─TcpConfig.json
├─PPHeat
│  ├─AliasTableConfig.json
│  ├─Config.json
│  ├─StateConfig.json
│  └─TransExcute.json
```

NodeRed

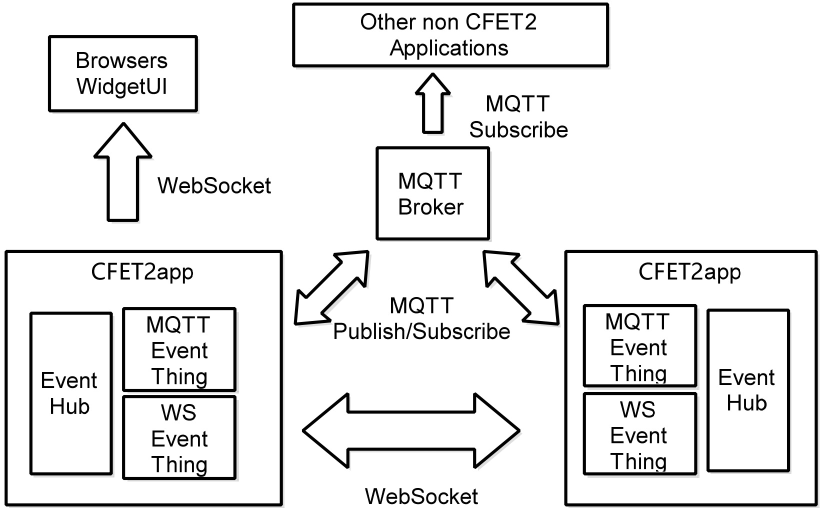
- **NodeRed for quick automation and HMI**
  - Control automation logic can be programed using drag & drop graphical interface
  - HMI can be developed using dash board





MQTT

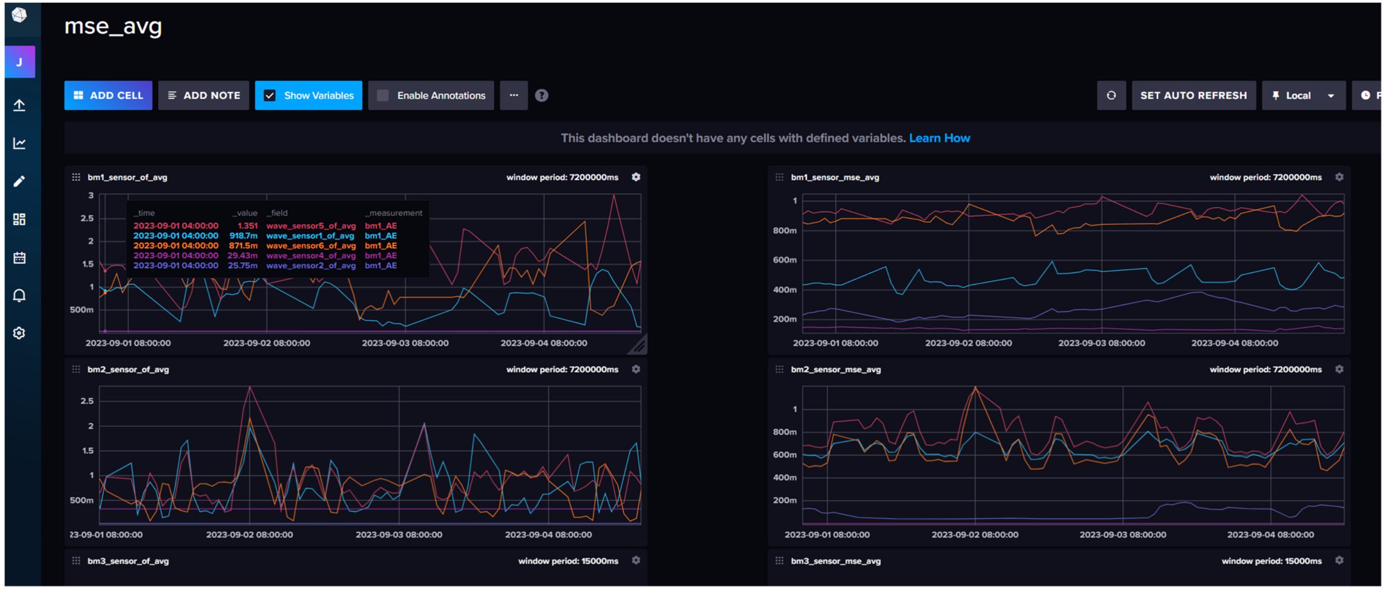
- **CFET2 support 2 event distribution:**
  - Web Socket for browser based client like the WidgetUI and everything else
  - MQTT for everything else



- Things inside CFET2app can use MQTT or WS for event, only difference is the URL of the event
- Non-CFET2app can just use MQTT to subscribe the event that’s the most simple way

InfluxDB

- **Using InfluxDB as archiver**
  - everything is HTTP so, just configure a Telegraf app to archive the resources
  - No need to make a new app

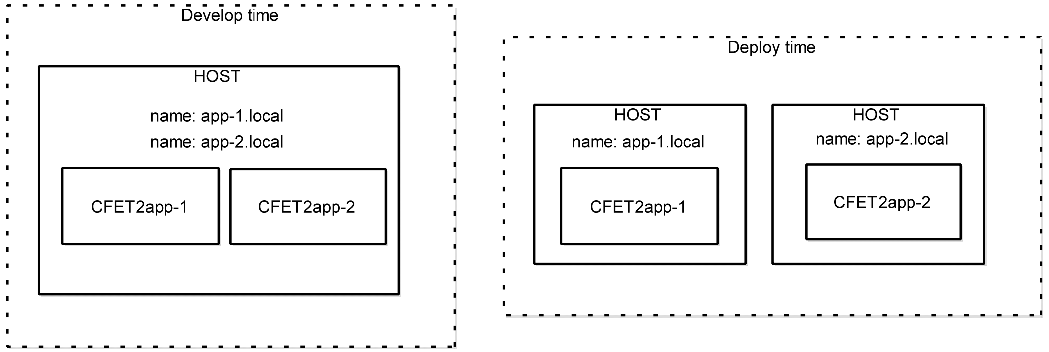


This dashboard doesn't have any cells with defined variables. Learn How

```
_time                _value   _field               _measurement
2023-09-01 04:00:00   1.351   wave_sensor5_of_avg  bm1_AE
2023-09-01 04:00:00  918.7m   wave_sensor1_of_avg  bm1_AE
2023-09-01 04:00:00  871.5m   wave_sensor6_of_avg  bm1_AE
2023-09-01 04:00:00  29.43m   wave_sensor4_of_avg  bm1_AE
2023-09-01 04:00:00  25.75m   wave_sensor2_of_avg  bm1_AE
```

mDNS

- **multicast DNS**
  - set host names so no ip address in the URL (.local domain names)
  - Using multiple domain names for the same host, each name for a CFET2app
  - Completely decouple CFET2app and CFET2 Resource from host
  - Just use Avahi

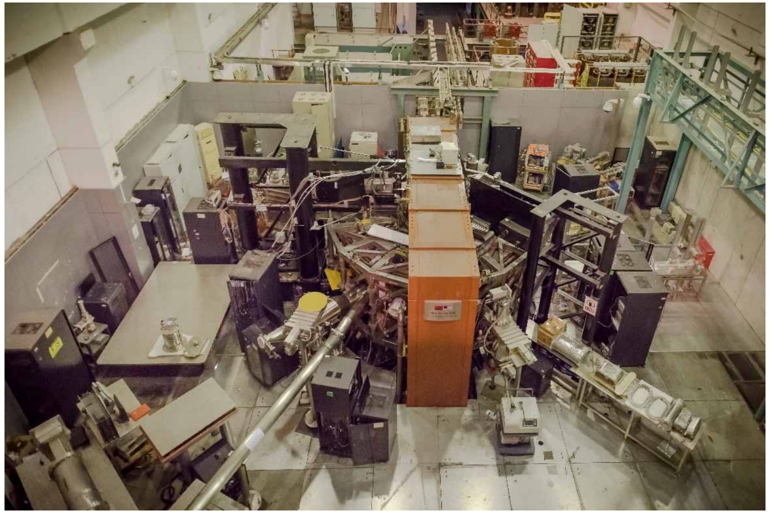


Outline

- Introduction on control system based on web technology
- A software toolkit for building a control system using web technology
- Leverage on existing technology enabling fast and easy control system implementation
- **Real world applications**

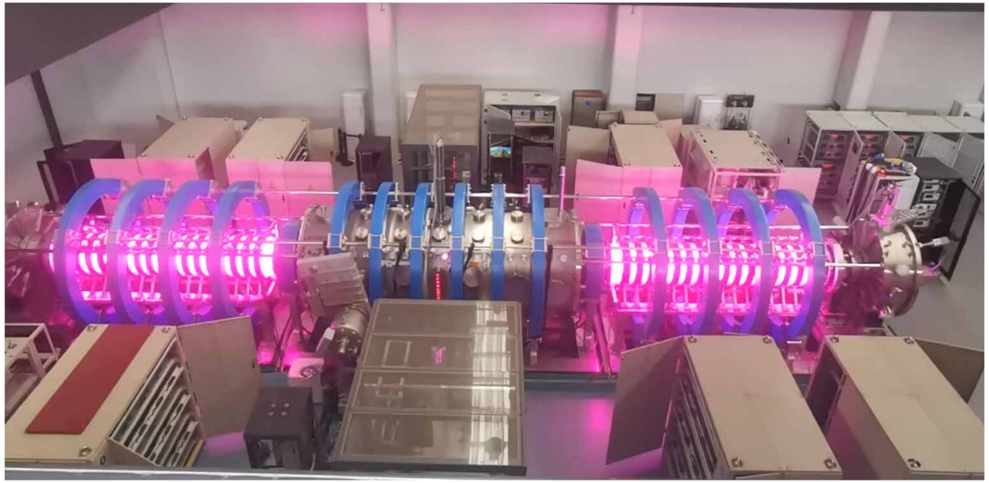
J-TEXT Tokamak

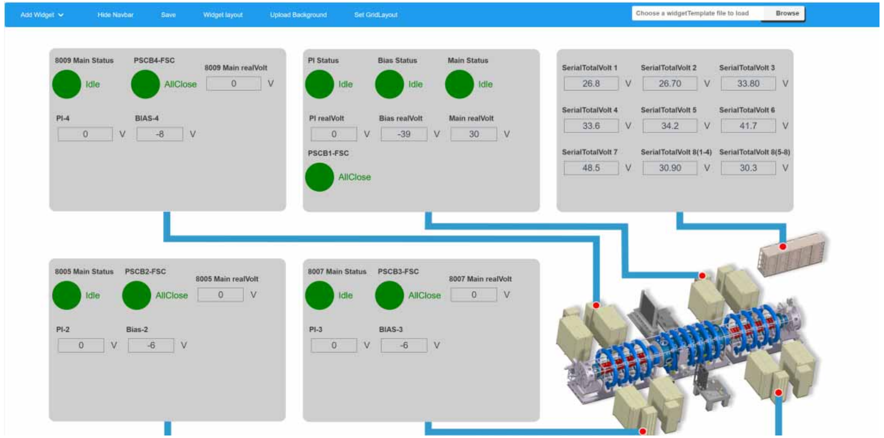
- **J-TEXT control system is based on EPICS CA**
  - Some of the new deployed systems are developed using CFET2
  - Works flawlessly with existing EPICS CA based systems.



HUST Field Reverse Configuration fusion experiment

- **China's largest FRC**
  - Everything is HTTP, even the alarm siren is HTTP (ESP32, HTTP over Wi-Fi)
  - Mostly powered by CFET2, but not necessarily everything, since HTTP is supported by everything





8009 Main Status ● Idle
PSCB4-FSC ● AllClose
8009 Main realVolt 0 V
PI-4 0 V
BIAS-4 -8 V

PI Status ● Idle
Bias Status ● Idle
Main Status ● Idle
PI realVolt 0 V
Bias realVolt -39 V
Main realVolt 30 V
PSCB1-FSC ● AllClose

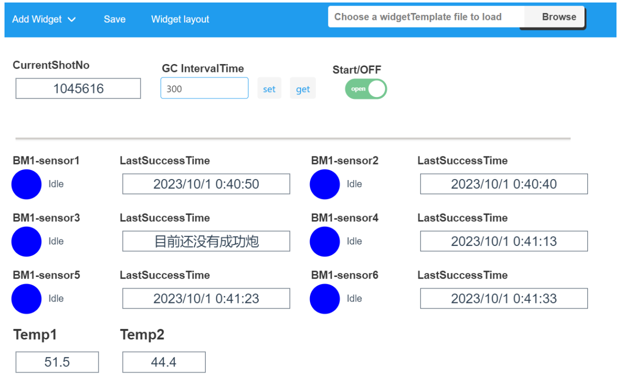
SerialTotalVolt 1 26.8 V
SerialTotalVolt 2 26.70 V
SerialTotalVolt 3 33.80 V
SerialTotalVolt 4 33.6 V
SerialTotalVolt 5 34.2 V
SerialTotalVolt 6 41.7 V
SerialTotalVolt 7 48.5 V
SerialTotalVolt 8(1-4) 30.90 V
SerialTotalVolt 8(5-8) 30.3 V

8005 Main Status ● Idle
PSCB2-FSC ● AllClose
8005 Main realVolt 0 V
PI-2 0 V
Bias-2 -6 V

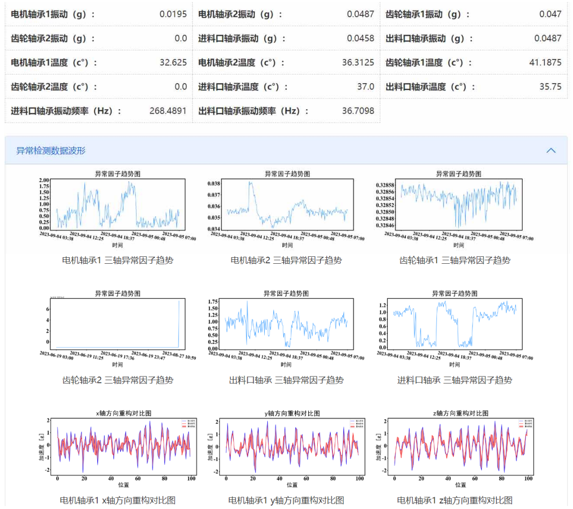
8007 Main Status ● Idle
PSCB3-FSC ● AllClose
8007 Main realVolt 0 V
PI-3 0 V
BIAS-3 -6 V

Smart Mining

- **Ball mill data acquisition and predictive maintenance**
  - Acquire vibration data using lots of sensor
  - Archive all the data
  - Using AI to predict anomaly and issue maintenance advices
  - Tested in on of the BAOWU's ball mill plant



CurrentShotNo	GC IntervalTime	Start/OFF	
1045616	300	open	
BM1-sensor1	LastSuccessTime	BM1-sensor2	LastSuccessTime
Idle	2023/10/1 0:40:50	Idle	2023/10/1 0:40:40
BM1-sensor3	LastSuccessTime	BM1-sensor4	LastSuccessTime
Idle	目前还没有成功炮	Idle	2023/10/1 0:41:13
BM1-sensor5	LastSuccessTime	BM1-sensor6	LastSuccessTime
Idle	2023/10/1 0:41:23	Idle	2023/10/1 0:41:33
Temp1	Temp2		
51.5	44.4		



电机轴承1振动（g）：	0.0195	电机轴承2振动（g）：	0.0487	齿轮轴承1振动（g）：	0.047
齿轮轴承2振动（g）：	0.0	进料口轴承振动（g）：	0.0458	出料口轴承振动（g）：	0.0487
电机轴承1温度（c°）：	32.625	电机轴承2温度（c°）：	36.3125	齿轮轴承1温度（c°）：	41.1875
齿轮轴承2温度（c°）：	0.0	进料口轴承温度（c°）：	37.0	出料口轴承温度（c°）：	35.75
进料口轴承振动频率（Hz）：	268.4891	出料口轴承振动频率（Hz）：	36.7098		

异常检测数据波形

电机轴承1 三轴异常因子趋势　电机轴承2 三轴异常因子趋势　齿轮轴承1 三轴异常因子趋势

齿轮轴承2 三轴异常因子趋势　出料口轴承 三轴异常因子趋势　进料口轴承 三轴异常因子趋势

电机轴承1 x轴方向重构对比图　电机轴承1 y轴方向重构对比图　电机轴承1 z轴方向重构对比图

BAOWU
宝武集团

World's largest steel producer

Summary

- Web technology can greatly improve the interoperability of the control system.
- With improved interoperability, mature solutions in other fields can be used in building control system for large experiments.

Thank you for you attentions!!