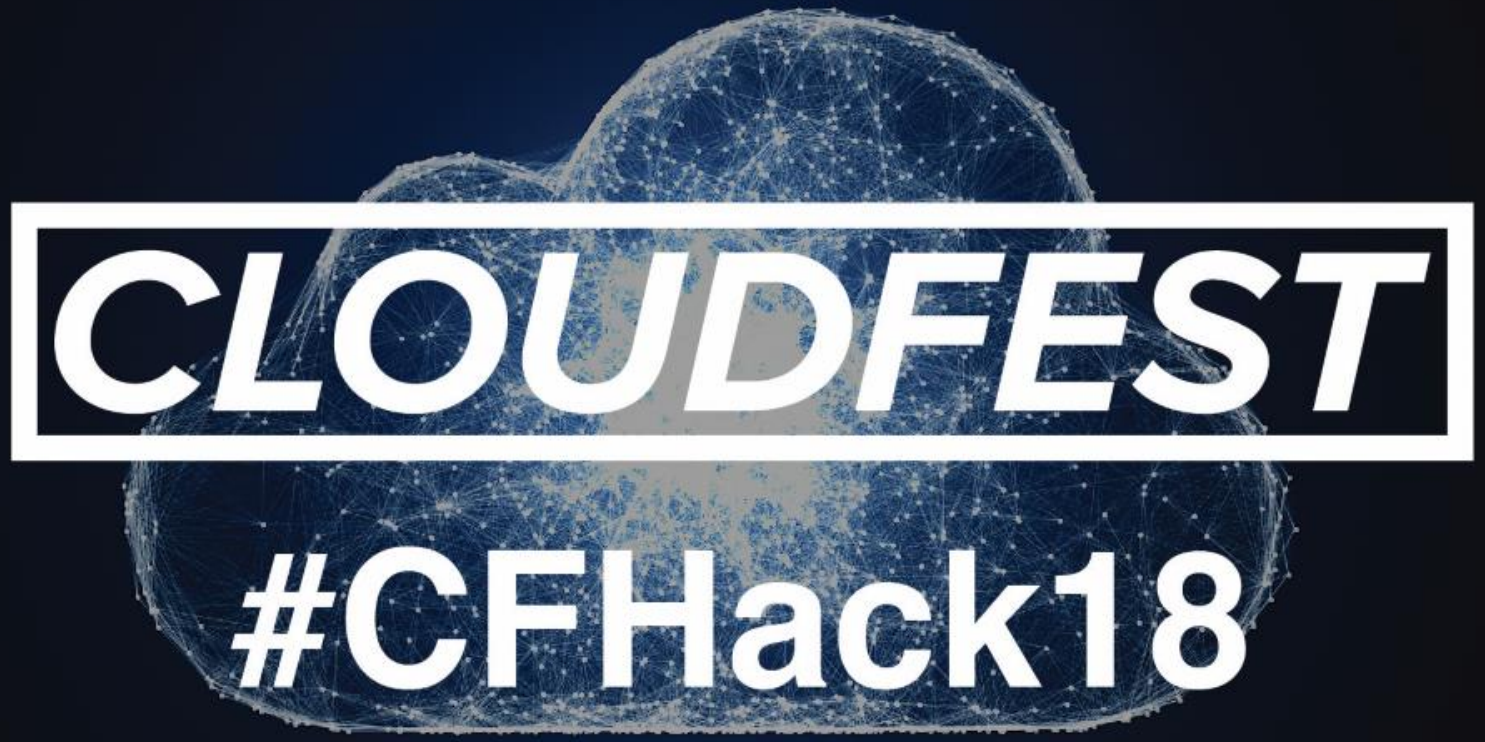

CLOUDFEST
#CFHack18


HEINRICH
I BUILT
I SECURED
I'M RUNNING
JOOMLADAY
JIDD
HACKAT
Partners
1&1
OVH

Project 1

Robert Windisch

Automated security check for WordPress plugins

Static Code Analysis

- Powered by RIPS Technologies
- High-tech company based in Bochum, Germany
- Supports the full feature stack of the PHP language
- Detects security vulnerabilities from user-controlled input
- Used by Open Source projects

RIPSTECH

SQL Injection

Write your content onto everybody else’s sites

CloudFest Hackathon - Expires on Mar 15, 2018, 11:59:59 PM

cloudfest

Summary | SQL Context

SQL Injection (unquoted)

The POST parameter 'dbname' is received in line 3 of the file [redacted] .php

The user supplied data is concatenated into sql markup in line 342 of the file [redacted] .php

The user supplied data is then used unsanitized in the sensitive operation mysqli_query() in line 343 of the file [redacted] .php. Please refer to the context and description for further information.

[redacted] .php

```
$_POST[ [redacted] ] $_POST[ [redacted] ] $_POST[ [redacted] ]
$_POST[ [redacted] ] $_POST[ [redacted] ] && $_POST[ [redacted] ]
```

Info | Desc. | Comments (0)

SQL Injection (unquoted)

Severity: High

OWASP Top 10:	A1	CWE:	89
SANS 25 Rank:	1	PCI DSS:	6.5.1

A sql injection vulnerability occurs when user input is embedded unsanitized into a SQL query. An attacker can modify the SQL syntax and alter the query's target or result. This can lead to the retrieval of sensitive information from the database or to an attack against the underlying web server by using SQL file operations. An attacker can also elevate privileges if the SQL

Codeviewer | Browser | .php | class. [redacted] .php

```
331     case "empty":
332         //DROP DB TABLES
333         $drop_log = "Database already empty. Ready for install.";
334         $sql = "SHOW FULL TABLES WHERE Table_Type != 'VIEW'";
335         $found_tables = null;
336         if ($result = mysqli_query($dbh, $sql)) {
337             while ($row = mysqli_fetch_row($result)) {
338                 $found_tables[] = $row[0];
339             }
340             if (count($found_tables) > 0) {
341                 foreach ($found_tables as $table
342                     $sql = "DROP TABLE `{$_POST[ [redacted] ]}.{$table_name}`";
343                     if (!$result = mysqli_query($dbh, $sql)) {
344                         [redacted]::error(sprintf([redacted], $_POST['dbname']));
345                     }
346                 }
```

File Upload

Write your files onto everybody else’s servers

Info | Desc. | Comments (0)

File Write (PHP file)

Severity: Critical

OWASP Top 10:	A1	CWE:	96
SANS 25 Rank:	10	PCI DSS:	6.5.8

A code execution vulnerability occurs when user input is embedded unsanitized into a code evaluation. It allows an attacker to break out of the evaluated PHP syntax and to injection arbitrary PHP code. Common character sequences injected use the curly syntax {${phpinfo()}} that leads to a successful code execution even within double quoted values. It is recommended to avoid the

Codeviewer | Browser | [redacted].php

```
100                 }
101                 if ( is_writable( WP_PLUGIN_DIR ) ) {
102                     $slash = '/';
103                     if ( [redacted]S ) {
104                         $slash = '\\';
105                     }
106                     if ( ! file_exists( WP_PLUGIN_DIR . $slash . $folder ) ) {
107                         if ( mkdir( WP_PLUGIN_DIR . $slash . $folder ) ) {
108                             $content = "<?php\n/[redacted]" . $_POST['[redacted]'] . "\n*/";
109                             if ( file_put_contents( WP_PLUGIN_DIR . $slash . $folder . $slash . $file, $content ) ) {
110                                 wp_redirect( admin_url() . 'plugins.php?page=[redacted]n=success&file=' . $folder . $slash . $file );
111                                 exit;
112                             }
113                             else {
114                                 wp_redirect( admin_url() . 'plugins.php?page=[redacted]b=true' );
```

Code execution

Run your code directly

Issue | Summary | PHP Context | Not reviewed

Info | Desc. | Comments (0)

Code Execution

Severity: Critical

OWASP Top 10:	A1	CWE:	95
SANS 25 Rank:	18	PCI DSS:	6.5.1

A code execution vulnerability occurs when user input is embedded unsanitized into a code evaluation. It allows an attacker to break out of the evaluated PHP syntax and to inject arbitrary PHP code. Common character sequences injected use the curly syntax {${phpinfo()}} that leads to a successful code execution even within double

Codeviewer | Browser | class.php

```
368        $start_time = microtime (true);
369
370        $php_error = "";
371        ob_start (
372
373        try {
374          eval ("?>". $code . "<?php ");
375        } catch (Exception $e) {
376            $php_error = "PHP error in " .  _NAME . " code block ". ($obj->number == 0 ? '' : $obj->number . " - ") . $obj
->get_ad_name() . "<br />\n" .  $e->getMessage();
377        }
378
379        $processed_code = ob_get_clean ();
380
381        if (strpos ($processed_code, __FILE__) || $php_error != "") {
382
383          if (preg_match ("%( +) in "  FILE  "%"  strip_tags($processed_code)  $error_message))
```

What we have achieved

- Reviewed findings for many plugins
- Most Plugins are secure
- Contacted plugin authors with vulnerabilities
- Build a PHP tool to use the API for WordPress and other projects

Project 2

François Serman

The problem


login username:password
OK
FTPd
BAD!
login username:password
OK

A solution: OTP

Client

ProFTPD

Auth Provider

username:nana-cats

OTP required

Press Button

****

Password:********

GET /auth/...

Status OK

AWESOME!!!!

Video demo

Done:

- Dockerised a ProFTPD build and run environment
- Modified mod_auth_otp to add Yubikey OTP validation
- Dockerised yubikeyedup for yubikey validation
- Used gitlab-ci and Rancher as devops pipeline
- Ate pizza, consumed lots of beer and coffee!

Containerise all the things!

TODO:

- Create a dedicated module for yubi OTP
- Allow for configuration of auth backend
- Collaborate with ProFTPD team for upstream integration

1&1 OVH Innovation is Freedom

Project 3

Michael Klein

Singed Autoupdate

A save way to deploy updates for developer

The Problem

- Online (auto) Updates are necessary for the maintenance of Web Software and Extensions
- Dealing with outdated software is therefore important but comes with its own problems
- If an update server gets compromised a large number of websites get infected

Our Solution

Sign Update

- We create a list with all file hashes of the update
- We sign our list with a private key and send it with our update package

Verify the Update on Installation

- We Unpack the update and check with a public key if the file list was from the developer
- We check each file against the hash list and the amount of files
- We discard the update if anything doesn‘t match

Toolset for Developer

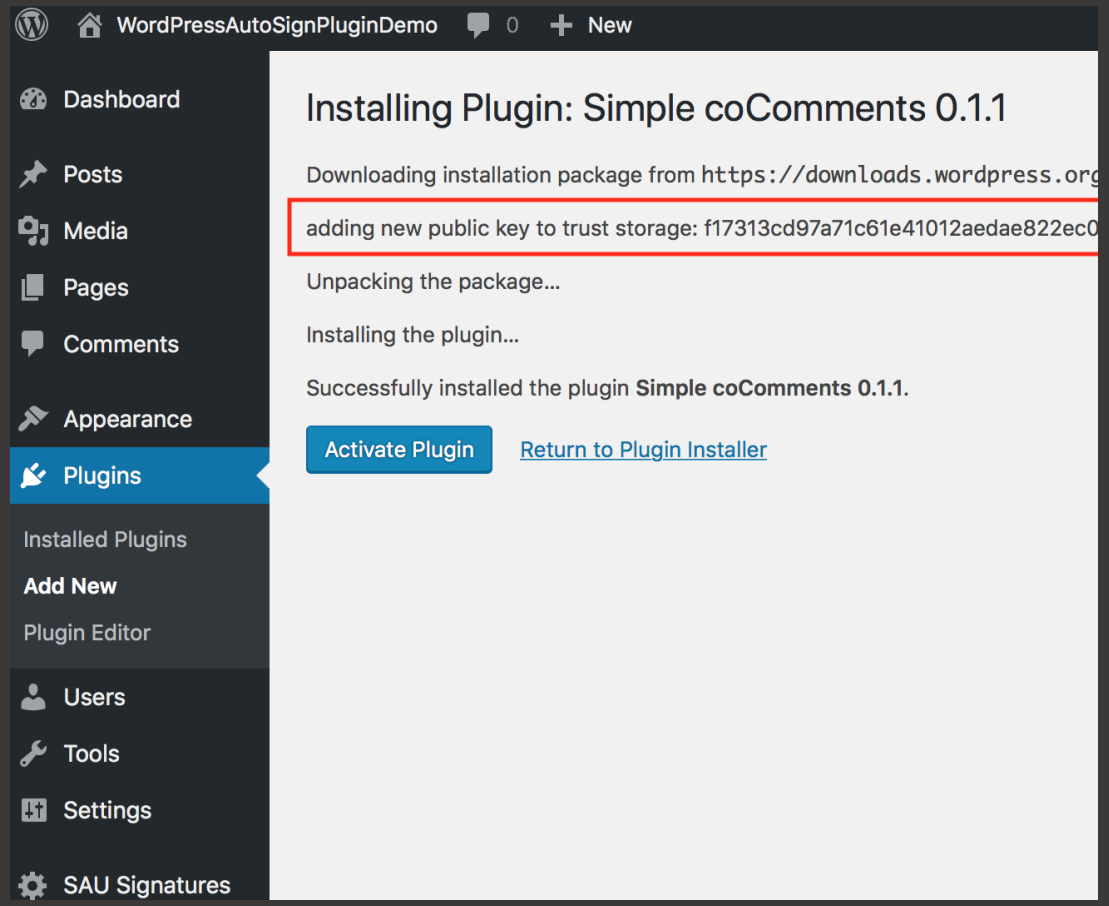
- CLI Tool for creating the Update with

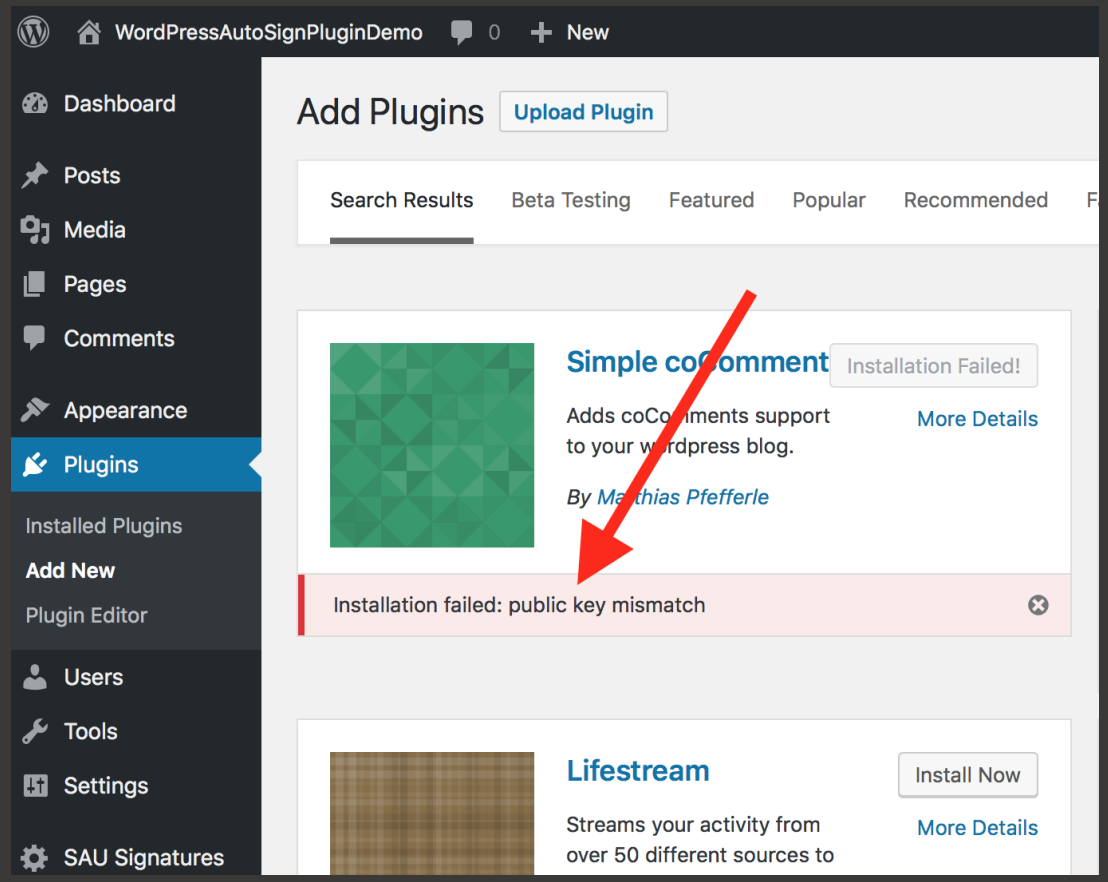
- $ signer.phar signer:sign [options] [--] <path> <key>

```
$public_key = hex2bin('< Developer Public Key >');

$update = new Update(__DIR__.'/update-deploy',$public_key);
$update->setTempDir('upload_test'); //optional
$update->ProcessUpdate('https://example.com/update.zip');
```

Wordpress Demo Plugin





GitHub

https://github.com/Cloudfest/signed-autoupdate

Project 4

David Jardin

Secure Websites and Content Management Systems

cms garden
eco VERBAND DER INTERNETWIRTSCHAFT
HACKMANIT
RUHR UNIVERSITÄT BOCHUM RUB
IT-Sicherheit IN DER WIRTSCHAFT
Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

https://www.joomla.de

Neuen Scan starten | Zeige Ergebnisse | Domain entfernen

Gesamtergebnis

Letzter Scan
10.03.2018 17:48

Header Scanner Mehr Informationen >>

Letzter Scan: 10.03.2018 17:48

- ✗ Überprüfung der Content Security Policy (CSP)
- ✓ Überprüfung des Content-Header
- ✗ Überprüfung des Public Key Pinning (HPKP)
- ✓ Überprüfung des HSTS Schutzes
- ✓ Überprüfung des X-Content-Type Headers
- ✓ Überprüfung der X-Frame Optionen
- ✓ Überprüfung des Cross-Site-Scripting Filters

DOMXSS Scanner Mehr Informationen >>

Letzter Scan: 10.03.2018 17:48

- ✓ Überprüfung des JavaScripts
- ✗ Überprüfung der Eingabe-Einstellung der Webseite

Joomla!

SAMPLE DATA

Blog Sample data

Sample data which will set up a blog site.
If the site is multilingual, the data will be tagged to the active backend language.

SIWECOS

Scan Results

- 90 / 100 Header Scanner
- 100 / 100 DOMXSS Scanner
- 90 / 100 Info Leak Scanner
- 100 / 100 Initiative-S Scanner
- 94 / 100 TLS Scanner

LOGGED-IN USERS

Super User Administration	2018-03-11 14:39
Super User Administration	2018-03-11 14:40

Incident Details

Created at 2015-10-22 16:00:00

CMS Joomla

Affected Versions 3.2.0 - 3.4.4

Exploittype SQLi

Description There is an SQL Injection vulnerability in Joomla versions between 3.2.x and 3.4.4 that allows unauthenticated attackers to read arbitrary data from the database, which enables them to gain full access to the site under some circumstances.

Can be filtered? Yes

Filter Description The vulnerability can be filtered with a web application firewall of your choice.

mod_security Rules

```
SecRule REQUEST_FILENAME "(index\.php|\/$)" "chain,id:002367,t:lowercase"
        SecRule REQUEST_METHOD "^(GET|POST|HEAD)$" chain
        SecRule ARGS:option "^com_" chain
        SecRule ARGS:/list\[.+\]/ "(union|select|drop|from|all|delete|where|like|into|outfil
```

```
SecRule REQUEST_URI "component\/" "chain,id:002368,t:lowercase"
        SecRule REQUEST_METHOD "^(GET|POST|HEAD)$" chain
        SecRule ARGS:/list\[.+\]/ "(union|select|drop|from|all|delete|where|like|into|outfil
```

Plaintext Rules

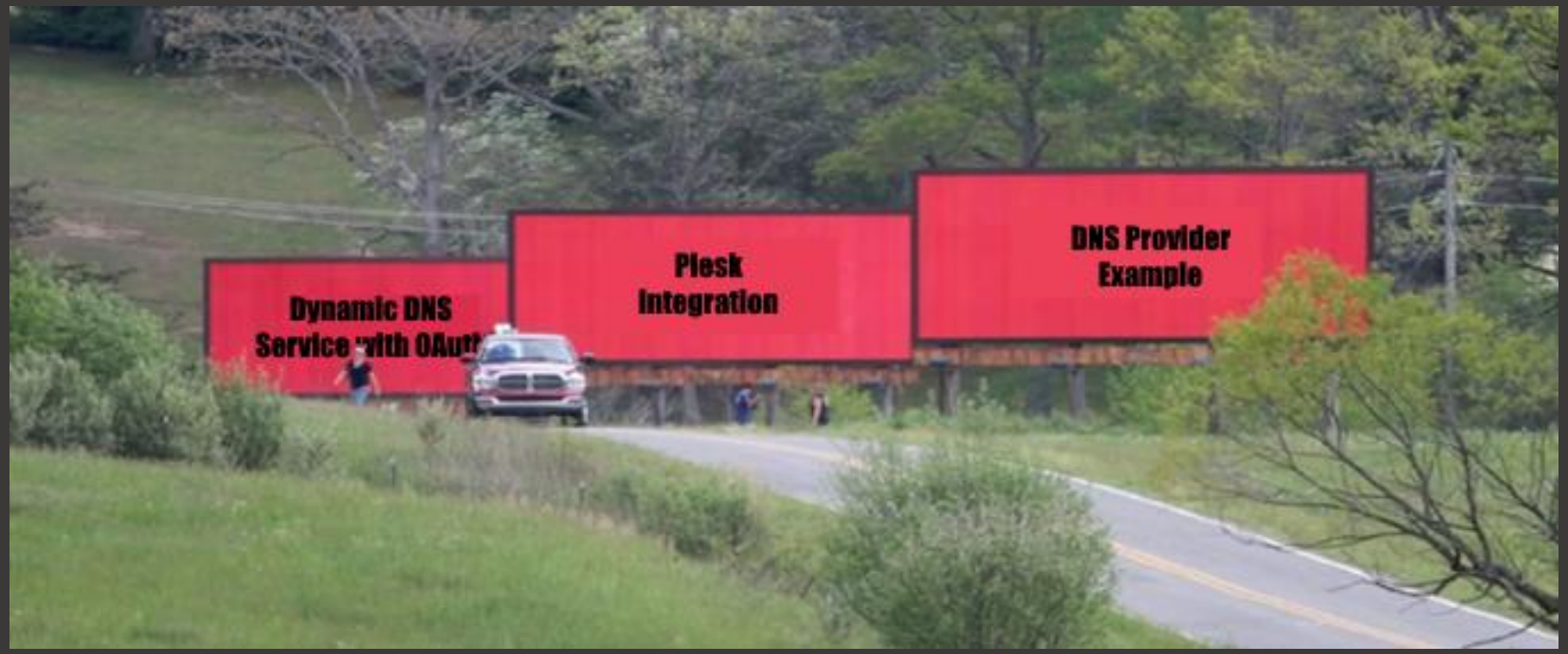
```
filter all requests that
have an index.php in their filename OR which's filename ends with a /
AND are GET, POST or HEAD requests
AND have a query parameter or POST payload for the option argument, that matches com_content
AND have a query parameter or POST payload for the view argument, that matches history
```

Project 5

Arnold Blinn

Domain Connect

Three Projects Outside of Rust, Germany



What is Domain Connect?

- Domain Connect is an open standard that makes it easy for a user to configure DNS for a domain running at a DNS provider to work with a Service running at an independent Service Provider. The user can do so without understanding any of the complexities of DNS.

- Supported by 20+ Service Providers, 14+ DNS Providers
  - Microsoft, Automatic, GoDaddy, 1&1, etc.

- http://domainconnect.org

Project 1: Example DNS Provider

- Goal: Build an Open Source Reference Implementation of Domain Connect for DNS Providers
- Challenge: Harder than the Service Provider Example (Requires State, and Working DNS)
- Components (all dockerized):
  - MySQL: Stores Users and Zones
  - DNS Server: Based on Open Source DNS, modified to work on MySQL
  - API Server: Implements Domain Connect API
  - Front End: Implements Domain Connect UX

Project 2: Plesk Integration

- Goal: Implement Domain Connect for DNS and Service Provider

- Plesk is a hosting control panel
  - Hosting
  - Email
  - DNS “Optional”

- Implementation
  - DNS Provider: When running DNS
    - Useful for email Services (O365), hosting services on sub-domains (blogs etc.)
  - Service Provider: When not running DNS
    - Allows configuration of host, email, and sub-domains to work

Project 3: Dynamic DNS

- Goal: Use Domain Connect to implement Dynamic DNS

- Dynamic DNS
  - Keeps IP current when host has a dynamic IP address from ISP
  - Often built into routers or services running on the host
  - No universal way to handle between DNS Providers
    - DynDNS has a protocol that made its way into routers
    - Different DNS Providers have bespoke APIs

- Implementation:
  - Model DDNS as a template
  - Installer application gets Oauth consent
  - Windows Service checks IP and applies template as necessary

Results

- All three projects will require refinement, but shown to be viable and will be further developed
  - DNS Service Example code will be open sourced
  - Plesk integration finished and shipped
  - Dynamic DNS Application open sourced and shipped as a proof of concept (branded Domain Connect)
- Identified minor specification changes (improvements) to support several of these scenarios easier
- Improved clarity on several complex issues in specification

Project 6

Marcel Wagner &
Michael Sommerer

CSP Ready IoT Solution for SMB

Ali Kocal (Intel), Jessica Smith (1&1), Marcel Wagner (Intel), Ben Rösler (GzEvD), Gabrielle W. Poerwarwinata (Intel), Christian Buchwald (TÜV Rheinland), Steven Briscoe (Intel), Jamal El Youssefi (Intel), Elias Hackradt (GzEvD), Chris Mcadam (1&1), Michael Sommerer (IDI GmbH)

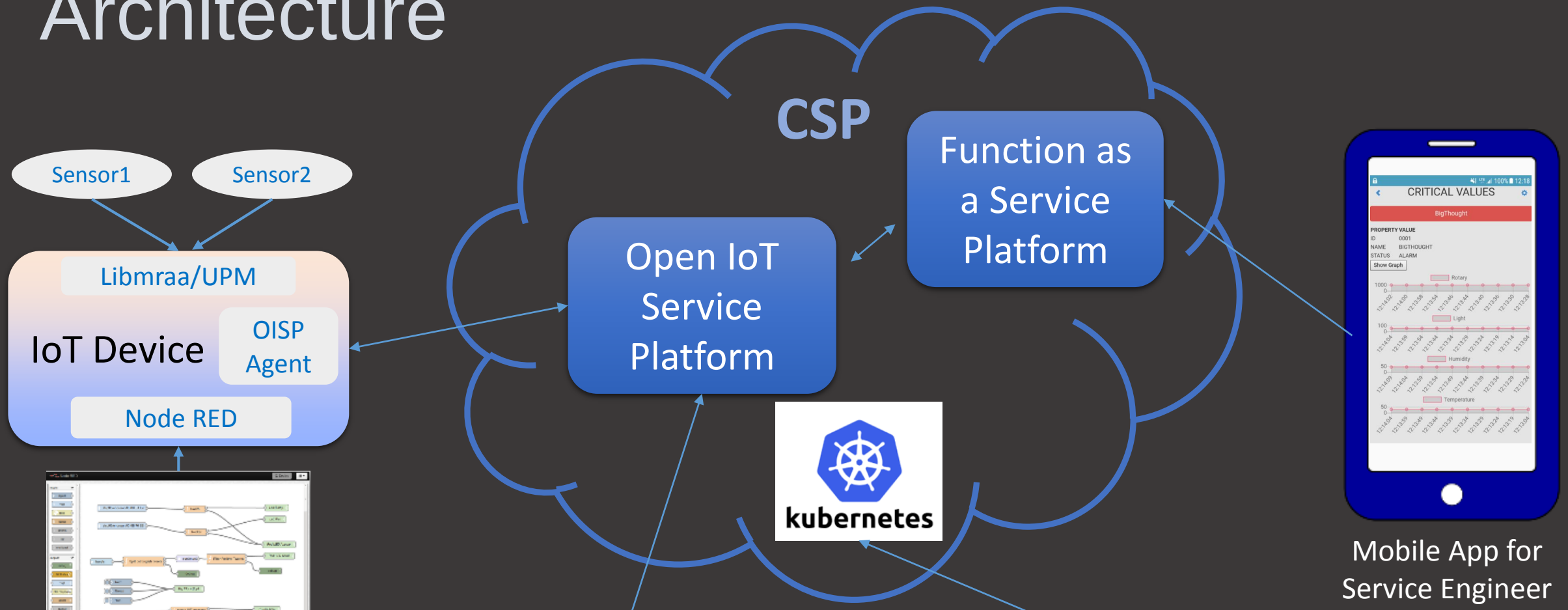
Problem Statement

- IoT Device integration with Cloud services is complicated and today based on proprietary solutions which have similar functionality but different API

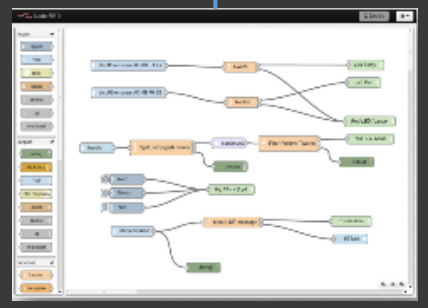
Target of this Project

- Develop an End to End Open Source architecture for CSPs and System Integrators ready to be deployed in Industrial environment
- Using last year’s Hackathon initiated Open IoT Service Platform (OISP) as middleware to orchestrate IoT devices and connect them with additional CSP Services

Architecture



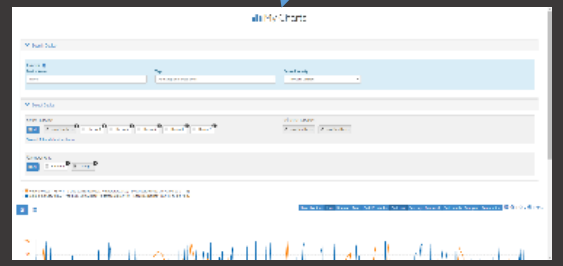
Mobile App for Service Engineer



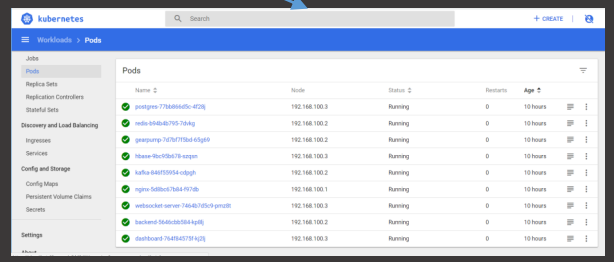
Node-RED GUI

<u>Hardware:</u>

UP Squared Grove IoT KitRaspberry Pi ZeroW

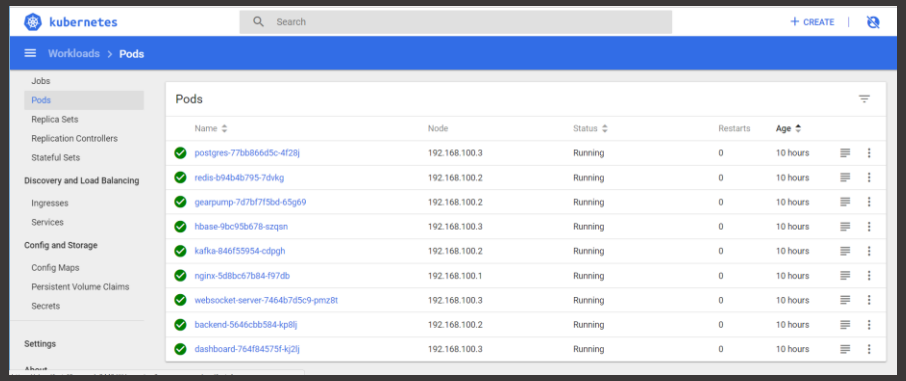


Dashboard/Admin GUI for OISP

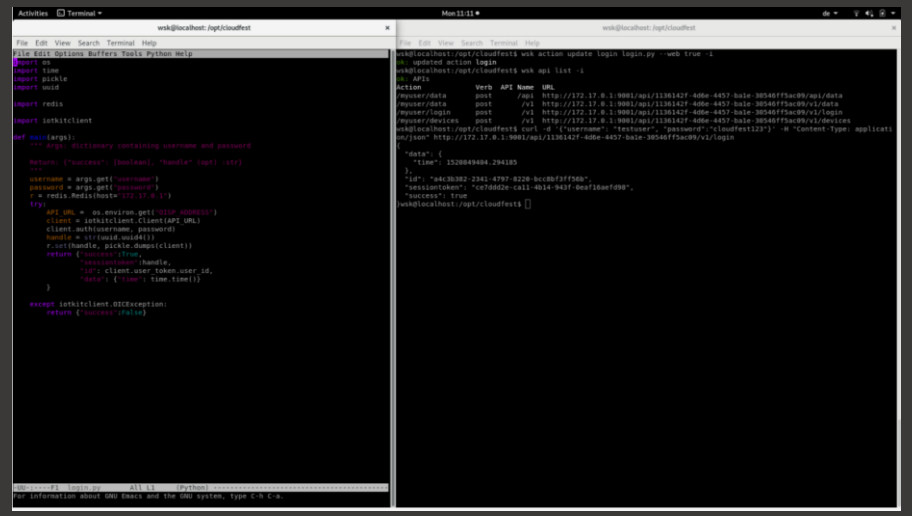


Kubernetes GUI

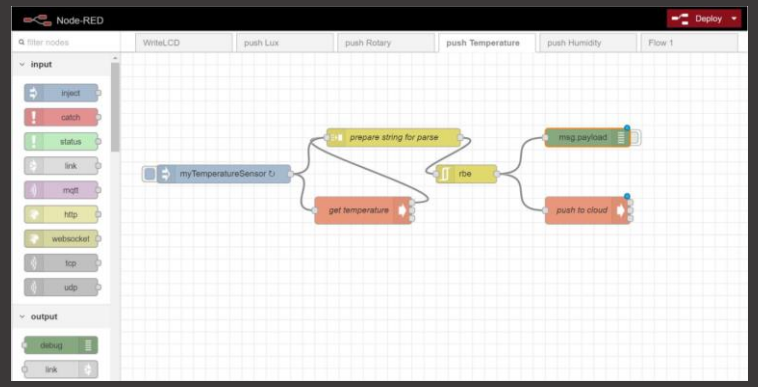
Impressions



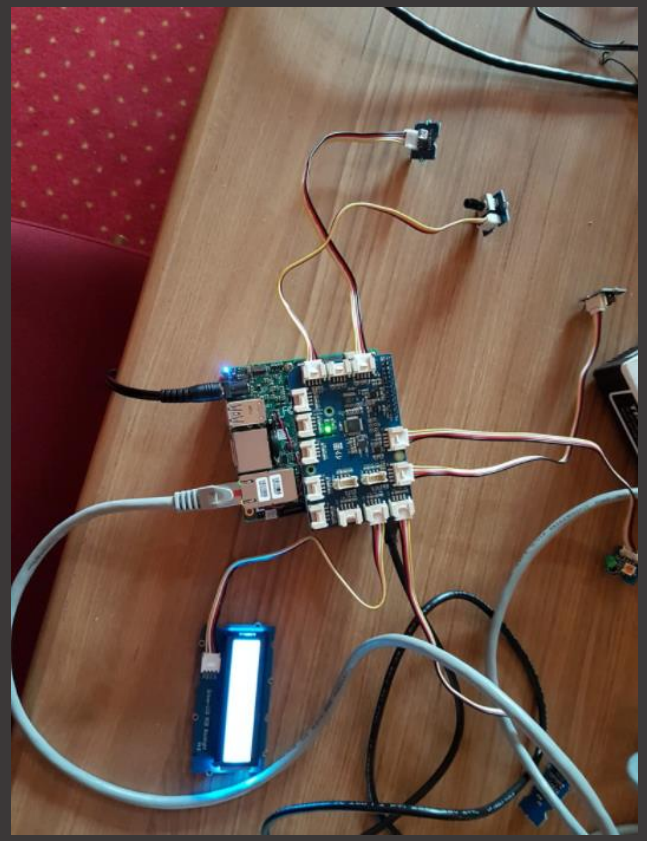
Kubernetes UI for OISP deployment

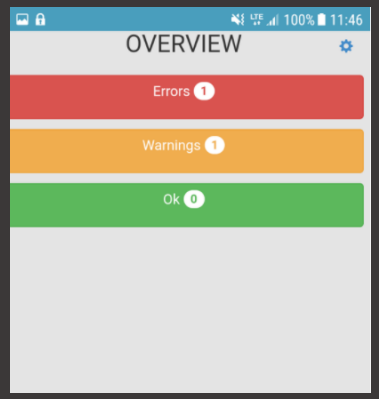


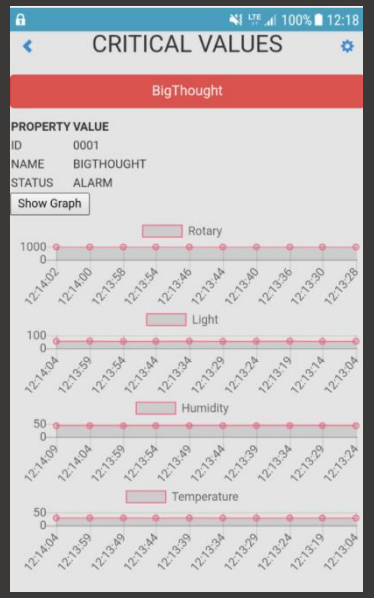
FaaS console to submit function



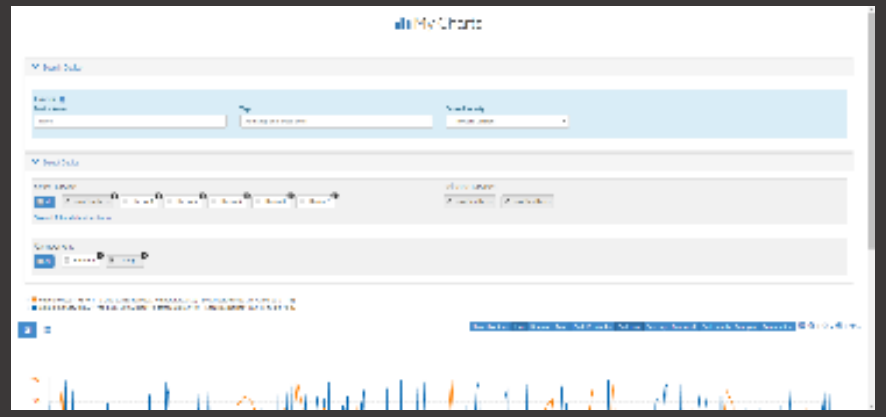
Node RED IoT configuration







Mobile App for Service Engineer



Service/Admin GUI

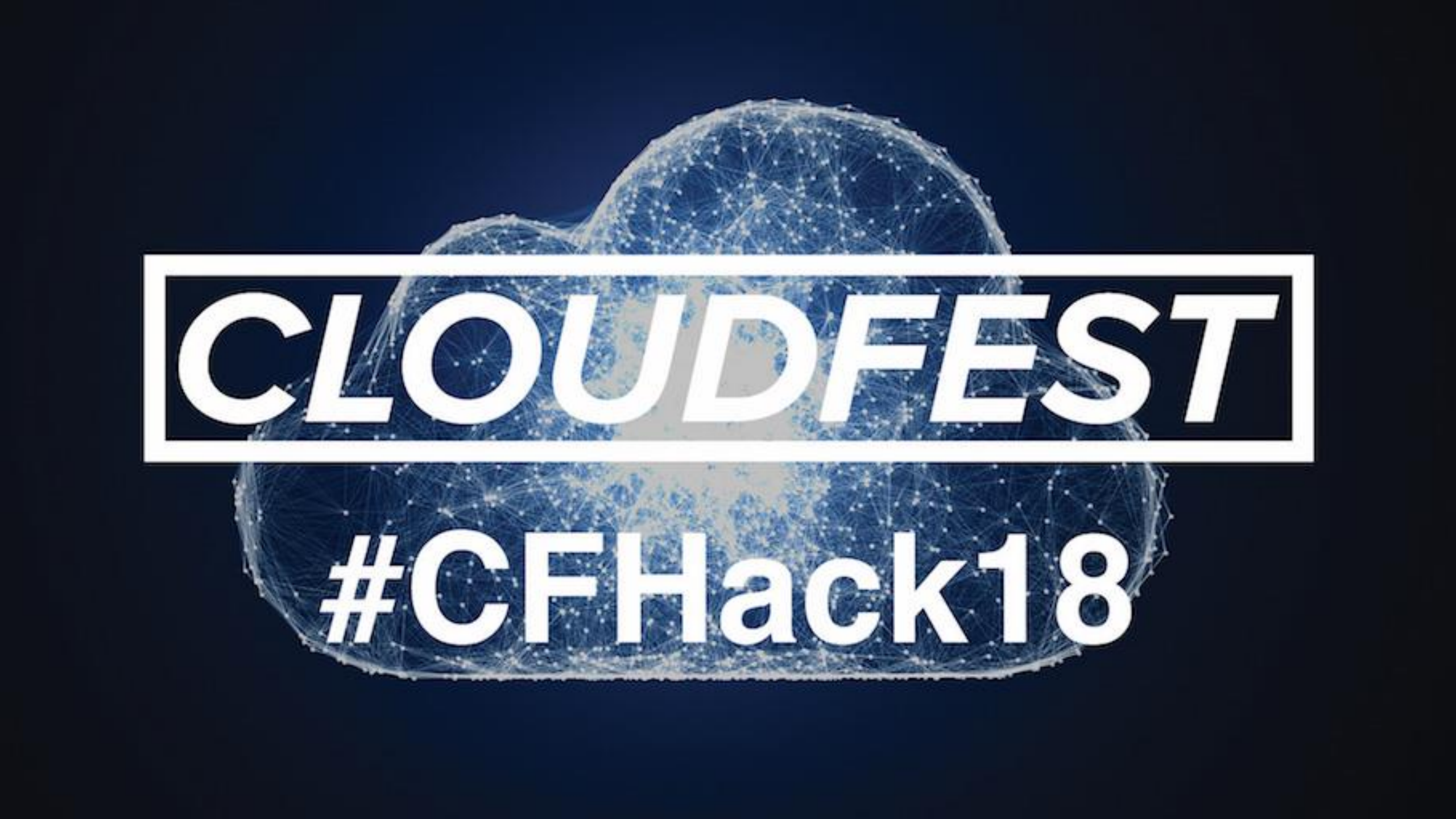
Results

During the Hackathon (2 days) we

- Decoupled IoT and Cloud dependencies by OISP services allowing efficient parallel development (IoT, Cloud and Mobile)
- Integrated Node RED with OISP on IoT Devices
- Made OISP deployable in CSP infrastructure with Kubernetes
- Integrated a FaaS framework (OpenWhisk) with OISP
- Developed a mobile application for local service engineer

- ALL Open Source and on github:
  https://github.com/Open-IoT-Service-Platform/platform-launcher

Our Hackathon Partners


CLOUDFEST
#CFHack18