State of the Celestial Sphere

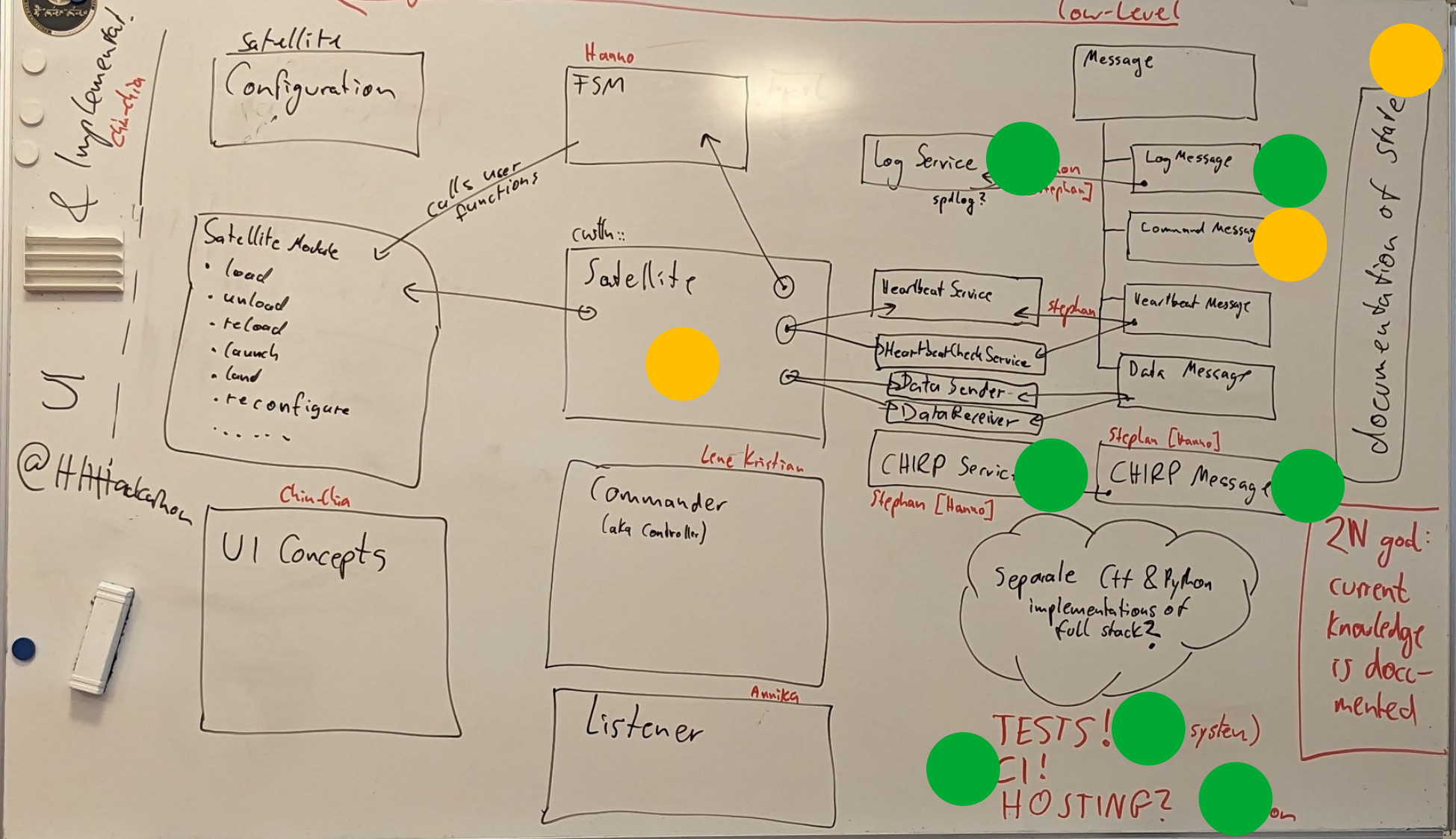
Ongoing Developments in Constellation

Simon Spannagel, DESY
EDDA Project Meeting
05/02/2024

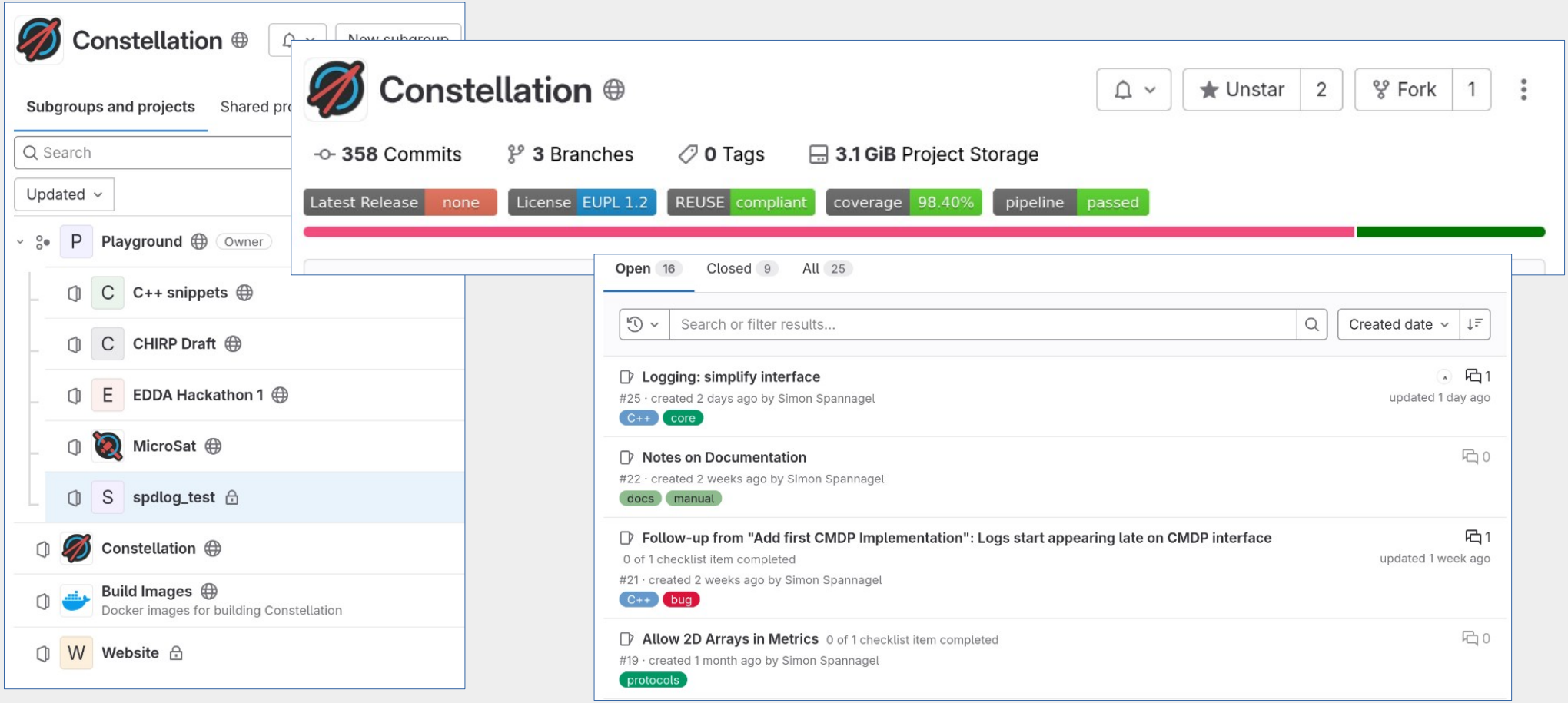
Overview

- Many things developed over past months
- Most infrastructure things solved, repositories on DESY Gitlab in action
- CI set up and running
- Implementation for micro controllers (ESPx)

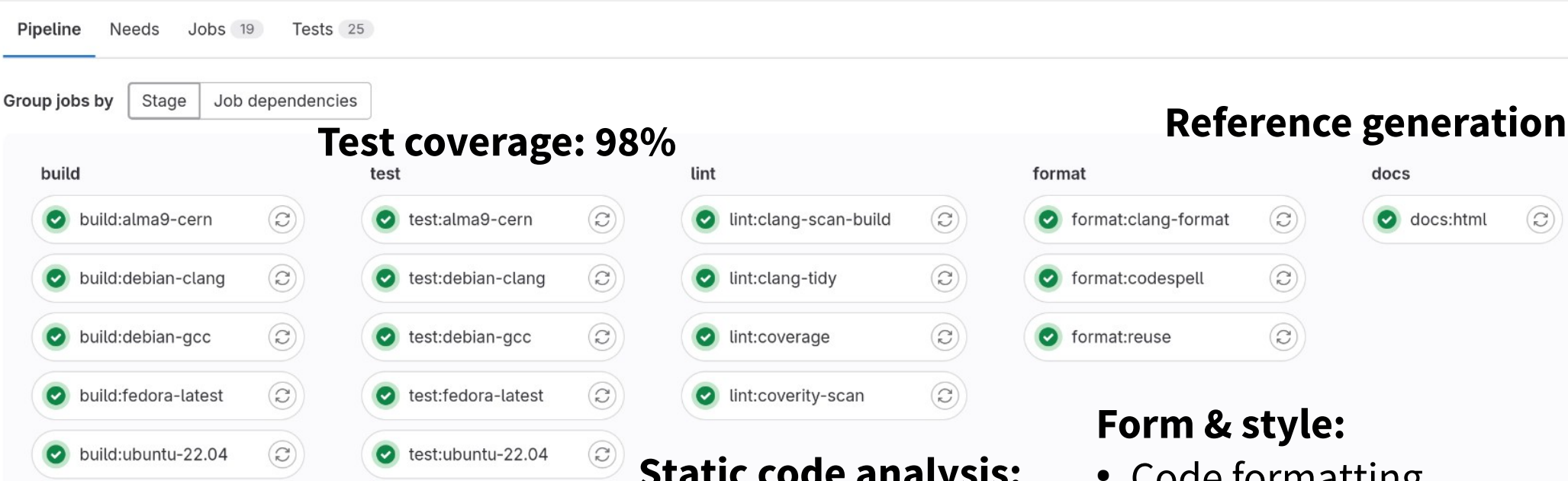
- Stephan is back! – working @ DESY since last week


Satellite
Configuration
Hanno
FSM
calls user functions
Satellite Module
• load
• unload
• reload
• launch
• land
• reconfigure
cwtfh::
Satellite
Message
Log Service
spdlog?
Log Message
Command Messag
Heartbeat Service
stephan
Heartbeat Message
HeartbeatCheck Service
Data Sender
DataReceiver
Data Messag
CHIRP Servic
CHIRP Messag
Stephan [Hanno]
Stephan [Hanno]
Lene Kristian
Commander
(aka Controller)
Chin-Chia
UI Concepts
Annika
Listener
Separate C++ & Python implementations of full stack?
TESTS!
CI!
HOSTING?
system)
documentation of share
2N god: current knowledge is documented
UI
@HHHackathon

Repositories @ DESY GitLab



Continuous Integration



Test coverage: 98%

Reference generation

Supported OS:

- Alma Linux 9
- Debian Testing
- Fedora 39
- Ubuntu 22.04

Static code analysis:

- Coverity
- Clang Tidy
- Clang Scan-Build

Form & style:

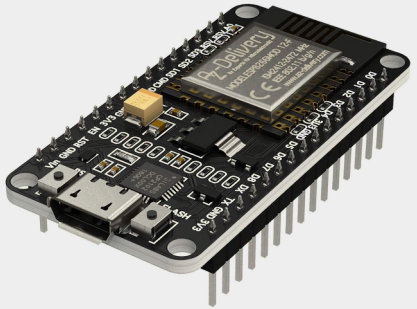
- Code formatting
- Spelling / typos
- Licensing checks

MicroSat – Constellation on Microcontrollers

- Had some ESP8266 lying around
  - Use them regularly for temperature/humidity monitoring
  - Thought: that would be cool to feed into Constellation

- Issue: no ZMQ implementation for microcontrollers (too heavy!)
  Solution: implement our own minimalist implementation
- Now have minimal **ZMTP** + pub-only **CMDP** + offer-only **CHIRP**!
  fully compliant with Constellation protocols

~3x6 cm
~5 EUR



MicroSat in Action

```
void setup() {
    // hostname of this machine, optionally a NTP
    CMDPPublisher::init(hostname, &timeClient);
    // Starting the publisher
    CMDPPublisher::begin();
}
```

```
void loop() {
    // Needs to be called on every loop iteration to process new clients and subscriptions/unsubscriptions:
    CMDPPublisher::update();

    // Logging a message:
    CMDPLOG(INFO, "This is a log message that will be distributed via CMDP");

    // Optionally, a logger topic can be set:
    CMDPLOG(WARNING, "SENSOR", "Something is up with the sensor reading");

    // For TRACE log levels, code location information is automatically attached
    CMDPLOG(TRACE, "There is a bug at this location that needs fixing");
}
```

```
[INFO] Connected to WiFi network MYWIFISSID
[INFO] To connect to CMDP, send a CHIRP request beacon to obtain the endpoint
[INFO] Alternatively, connect directly to  192.168.179.12 : 53080
[DEBUG] CHIRP: Preparing to send CHIRP message of type 2 for service 3
...
[DEBUG] ZMTP: Received connection from client at 192.168.179.3 : 39432
[DEBUG] ZMTP: Verified ZMTP signature match
[DEBUG] ZMTP: Verified ZMTP version match
[DEBUG] ZMTP: Verified that remote peer is SUB socket, accepting.
[INFO] CMDP: Accepted new connection from client 0 at 192.168.179.3 : 39432
[INFO] CMDP: Client 0 subscribed to topic:  LOG/WARNING
```

Next steps (for me)

- Pick up on writing conceptual design documentation

- Documentation will be divided into 4 parts:
  - Tutorials: teach how to use Constellation
  - Concepts: write up thoughts behind the framework structure
  - How-To Guides: concise answers how to achieve a specific goal
  - Reference: detailed documentation one can consult to get info on inner workings, parameters, etc
- See https://diataxis.fr/ for some theoretical background on this

Things we could tackle already

- Chinchia presented some tools & ideas for GUI design
- We should identify tools / frameworks in which we would like to implement Uis
  - Python: Textual (https://textual.textualize.io/)?
  - C++: Qt6? Azul? FTXUI?
- Check out and add your ideas to the issue: https://gitlab.desy.de/constellation/constellation/-/issues/13
- Start drawing some ideas so we can iterate?


DESY.