



# AGL Signaling

Month 11, 2017 | OpenXC, Cloud, OpenAPI



**Fulup Ar Foll**

*Lead Architect, IoT.bzh*



# Beyond CAN Signaling Agent



# Signaling Requirement @ AGL

- Clear Isolation

- Low level CAN operation only depends on equipment
- High level business logic dedicated to applications

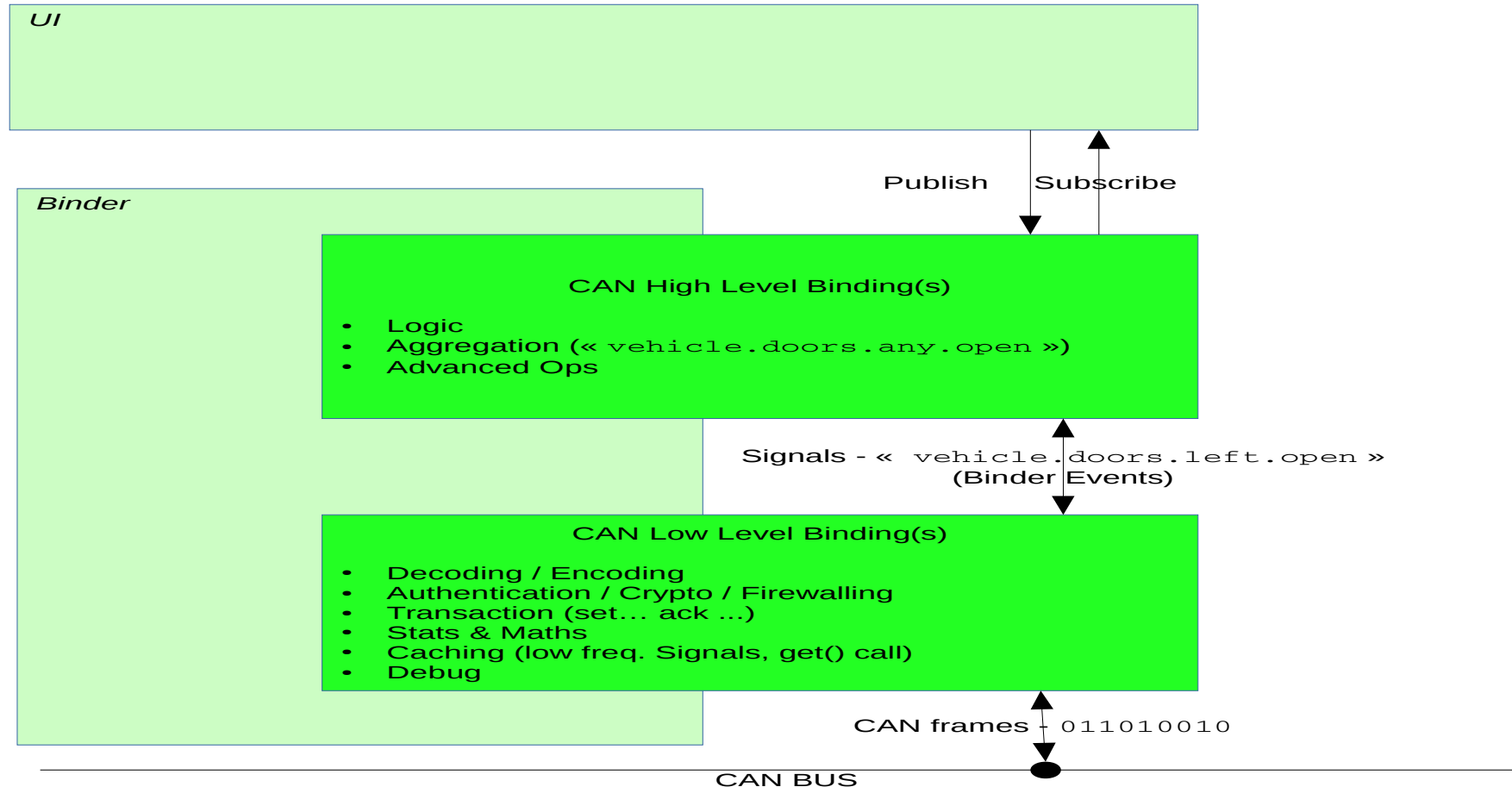
- Security Built In

- Navigation APP may access GeoLocation but not Telephony
- Implement statistic/counter to monitor unexpected behaviour
- Leverage AGL-CC framework

- API transparency for client applications

- Reuse existing technology (faster, cheaper, safer)

# Clear Isolation



# Low Level Binding

- Binary to Json
  - Binary encoding decoding
  - Generate Application Friendly Signal Name & Values
- Close to Automatic Code Generation
  - OpenXC CAN vector definition in JSON
  - Other CAN vector as CANoe
- Include Basic Filtering & Statistic
  - GT/GE LT/LE
  - Timer, Cycle, Timestamps
  - Counter: last value, average, invalid ID, ...
  - Can be ship to developer as binary only

# High Level Binding

- Provide stable API to applications
- Provide signal composition
- Can be split/overloaded to simplify security
- Should support vehicle to cloud
  
- As today investigating VIWI-W3C



# Security Model base of AGL AppFW

# Leveraging AppFW transport & security

```
schema-agl-api-v2.json  can-signal-api-v2.json  can-signal-api-v2.c x
1  static struct afb_auth auths[] = {
2      { .type = afb_auth_Permission, .text = "urn:AGL:permission:low-can:partner:read" },
3      { .type = afb_auth_Permission, .text = "urn:AGL:permission:low-can:partner:write" },
4      { .type = afb_auth_And, .first = &auths[0], .next = &auths[1] }
5  };
6
7  static const struct afb_verb_v2 verbs_v2[] = {
8      { .verb = "subscribe", .callback = f_subscribe, .auth = &auths[2], .session = AFB_SESSION_CHECK, },
9      { .verb = "unsubscribe", .callback = f_unsubscribe, .auth = &auths[2], .session = AFB_SESSION_CHECK, },
10     { .verb = NULL }
11 };
12
13  /* the integer data used by binder for the verbosity of the binder */
14  int afbBindingV2verbosity;
15
16  /* the structure for describing the binder */
17  const struct afb_binding_v2 afbBindingV2 = {
18     .api = "low-can", .specification = /* the JSON description */ .verbs = verbs,
19     .init = NULL, .start = start_low_can, .onevent = NULL,
20 };
```



# OpenAPI as API definition tool

```
schema-agl-api-v2.json  can-signal-api-v2.json x
1
2  {
3    "openapi": "3.0.0",
4    "$schema": "file:~/.openapi/schema-agl-api-v2.json",
5    "info": {
6      "description": "Can Signal Low Level API", "title": "low-can", "version": "2.0"
7    },
8    "servers": [{
9      "url": "ws://{host}:{port}/api/low-can", "description": "The API server.",
10     "variables": { "host": { "default": "localhost" }, "port": {"default": "1234"}
11   },
12   "X-afb-events": [
13     { "$ref": "#/components/schemas/afb-event" }
14   ]
15   },
16 ],
17 "components": {
18   "schemas": {
19     "afb-reply": {
20       "properties": {
21         "jtype": {
22           "type": "string"
23         },
24         "request": {
25           "$ref": "#/components/schemas/afb-request"
26         },
27         "response": {
28           "type": "object"
29         }
30       }
31     }
32   }
33 }
```

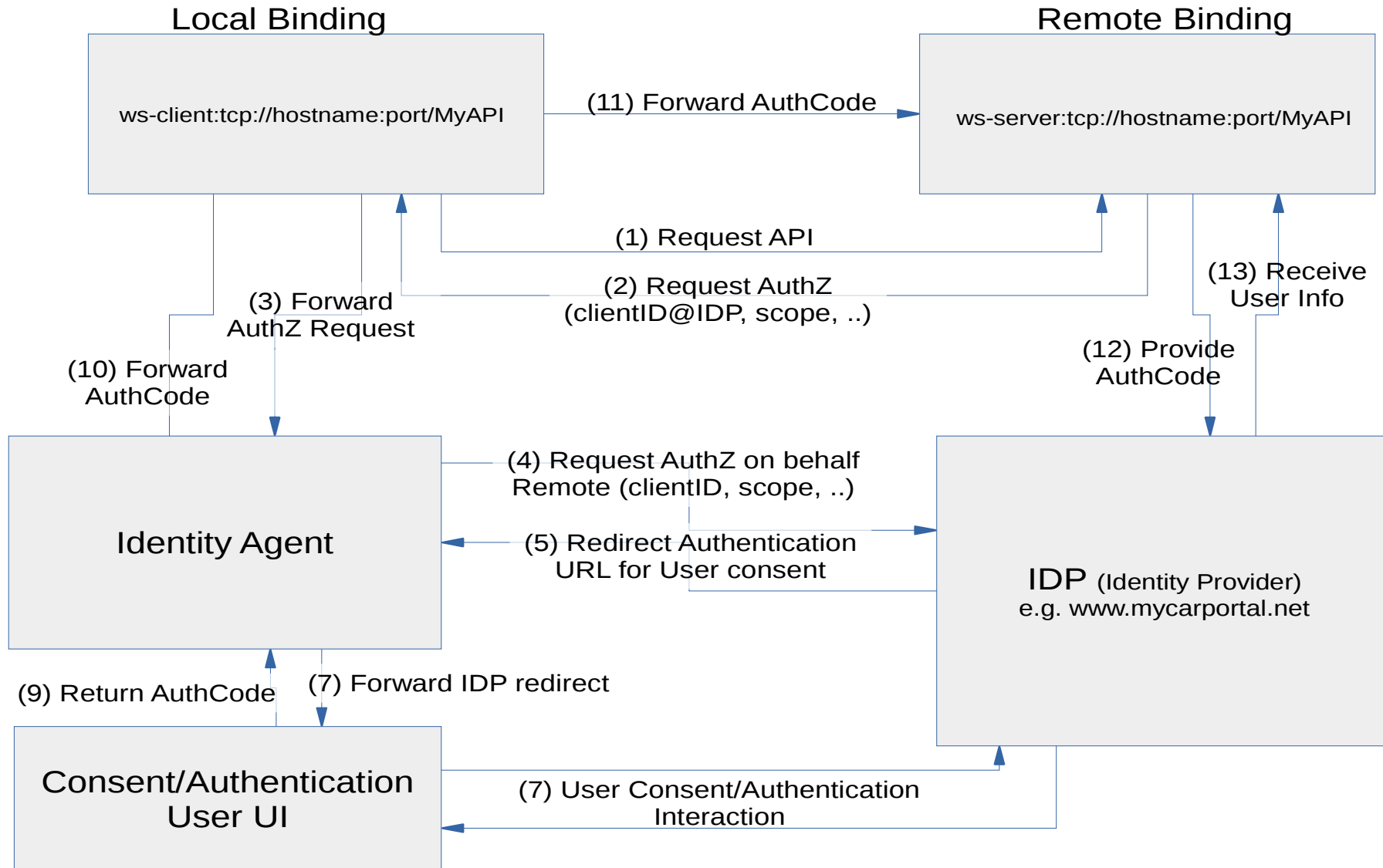
# Monitoring & Debug

- AGL/Master (available from git)
  - Binder Introspection (global or by API)
  - Verbosity change (global or by API)
- AGL/DD (June 2017)
  - API trace (somehow equivalent to tcpdump)
  - Application Log trapping
  - Asynchronous event observation
  - Integration with existing debug tool (ie: wireshark)
- AGL/EE (December 2017)
  - GDB integration
  - Client session tracing through API architecture
  - Central daemon to allow binder discovery, introspection & tracing

# Vehicle to Cloud



# Signal integration with V2C (AGL/2018)



# Thank you!

Visit GENIVI at <http://www.genivi.org> or <http://projects.genivi.org>

Contact us: [help@genivi.org](mailto:help@genivi.org)

## Further information

- Publication at <http://iot.bzh/publications>
- Documentation <http://docs.automotivelinux.org>

This work is licensed under a Creative Commons Attribution-Share Alike 4.0 (CC BY-SA 4.0)  
GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries.  
Copyright © GENIVI Alliance 2017.

