SERVICE ORIENTED ARCHITECTURE IN AUTOMOTIVE

IMPACTS AND CHALLENGES OF AN ARCHITECTURE PARADIGM CHANGE

RENAULT NISSAN MITSUBISHI

Authors:

Marc BELLANGER and Edward MARMOUNIER

(Renault Software Labs, Communication & Vehicle Information)

INTRODUCTION

The challenge to introduce SOA in automotive can be assessed through multiple questions :

- Main SOA concepts to adopt, question or ignore in the context of automotive?
- What impacts on the organization and processes of an automotive company?
- How to assess the actual benefit in term of time to market improvement for new features?
- How to guarantee the same level of safety and reliability of the overall system?

MAIN CONCEPTS ADOPTION: CONTRACT BASED SOFTWARE (1/2)

- Helps defining responsibilities split between actors
- Focus on the SW business of the contract first
- Formal definition of the contract
  - Architect / designer: model based design. UML?
  - Developer / machine : IDL with toolchain for interface code generation and verification
  - Strong typing for business operations!
  - Unambiguous namings
  - Static view and dynamic sequences
  - Share the contract: deploy it on a common space where client and server sides can consume / honor it

MAIN CONCEPTS ADOPTION: CONTRACT BASED SOFTWARE (2/2)

- ➢ Versioning the contract = preparing future evolution of the SW
- ➢ Enforce stable API between SW components:
  - Semantic versioning MAJOR.MINOR.PATCH (https://semver.org/)
  - Version of a contract supported by a service and by a client defines the SW life-cycle strategy (monolythic vs modules)
  - Components version vs contract version must be tracked (dependency graph)
- ➢ Track history of each component version for each vehicle to allow the maintenance of compatibility matrix at design time

MAIN CONCEPTS ADOPTION: TRANSPORT ABSTRACTION

- SOA frameworks generally offer abstraction over transport (IP, Unix-domain, SHM, proprietary IPC)
- What about latency? Automotive grade ethernet is a trend for inter-ECU SOA for non-realtime services.
- SOA latency constraints shall be expressed at interface modeling time and collected by the network team that guaranties the QoS (802.1Q/P mainly).
- SW developer focuses on the business of the contract, not how it is transported on the wire.
- This is a major change compared to signal-based legacy definition (CAN).

MAIN CONCEPTS QUESTION: DYNAMIC SERVICES DEPLOYMENT

Service Discovery:

- locating service instances at runtime, not adopted to avoid service spoofing and reduce service location delays. Static configuration may be more appropriate for services where startup latency is critical.
- detection of the availability of a service, adopted for a basic level of health monitoring.
- publish & subscribe relying on SD, adopted for notifications from service to client.

SERVICE ORIENTED ARCHITECTURE IN AUTOMOTIVE

SELECTING THE RIGHT SOA FRAMEWORK FOR AUTOMOTIVE (1/3)

Discovery
Language independent definition
Transport abstraction, serialization abstraction
Performance
Local / external comm
Security
Open source activity
Licensing
Languages support
Linux
QNX
Android
AutoSAR™

SELECTING THE RIGHT SOA FRAMEWORK FOR AUTOMOTIVE (2/3)

- Car ECUs don’t share the same eco-systems. Ex: Android IVI, QNX cluster, homemade Linux on IVC, AutoSAR™ for other ECUs.
- A standard would allow adopting SOA on car vehicle network, has it ever existed?
- Renault chose SOMEIP + CommonAPI, motivated by the feature coverage + adoption rate of SOMEIP in automotive industry.
- Renault participates to standardization consortium (AUTOSAR™) and/or open-source alliance (GENIVI GPRO) to expose the problem statement of SOA compatibility across multi-OS:
    - Franca usage guidelines and
    - IDL convertor tool (Franca <-> ARXML)

SELECTING THE RIGHT SOA FRAMEWORK FOR AUTOMOTIVE (3/3)

Android specific case

- SOMEIP based SOA does not fit in Android partitioning, layering and security patterns. Ex: unix domain sockets are prohibited between HAL layer and applicative layers.
- Horizontal and vertical inter-service communications shall stick to binder (AIDL / HIDL).
- Trend at Renault is to keep SOMEIP below the HAL and wrap services with Native or Java Car Services, re-exposing business APIs through AIDL.

IMPACTS ON THE ORGANIZATION AND PROCESSES: IMPROVING SEMANTIC DEFINITION

Architecture:

- ➢ Model-based SW, ensure consistency and improve detailed design
- ➢ How to represent the semantic of the service and its data parameters? Range, precision, units, asynchronous…
- ➢ How to specify safety and timing constraints in the model?
- ➢ UML is not expressive enough to carry such level of details
- ➔ Renault has developed internal UML extensions to annotate services definition, allowing:
  - explicit constraints directly in the model
  - easily extraction of network resources constraints to help Network teams deal with QoS

SOA IMPACTS ON ORGANIZATION: INTERNAL TEAMS (PROS)

- ➢ SW developers are enabled early in the development phase to run pre-integration tests of SW components with interfaces deployed on simple mocks (no need for CAN simulator), allowing parallel developments of Client / Service counterparts
- ➢ Helps splitting roles & responsibilities between SW and network
- ➢ No more boilerplate developments, straight to the business
- ➢ Versioning of the interface fits well to Agile SW increments
- ➢ SW reuse maximized across platforms:
  - services reuse from catalog…
  - …but mainly for clients where valuable business logic stands
  - reduces time to market on new platforms

SOA IMPACTS ON ORGANIZATION: INTERNAL TEAMS (CONS)

- It took time for developers to trust and learn SOA middleware and embrace the concepts
- It took time to network team to accept the roles split with SW definition
- Need to be careful not to forget performance considerations, optimization must be done some times on the interface itself (ex: Vehicle Signal Server with high throughput of data)
- SOA performances not proven versus signal based CAN, full IP car is not yet on the road :-)
- Some environments like Android break the “unique SOA standard” approach, requiring adaptations that have an R&D cost

SOA IMPACTS ON ORGANIZATION: PARTNERS (PROS)

- Definition of interfaces shall be a deliverable preliminary to any development
- No respect of the (static) contract = early toolchain error
- Integration camps: no more big bang, only dynamic behavior has to be verified
- Time to market and cost of integration are reduced for both Renault and partners
- Definitions are now SW contracts, no longer CAN message repository
- Once interfaces are defined, we register them into a common MBSW catalog where changes are ruled by a Change Control Board and semver principles.

SOA IMPACTS ON ORGANIZATION: PARTNERS (CONS)

- Partner must be aligned on the same SOA middleware
- Sharing SW contract done through IDL, not UML due to different modeling tools

REMAINING OPENS

Security:

- VSOMEIP implementation of SOMEIP standard proposes an access control level but only for intra-ECU.
- AUTOSAR™ ADAPTIVE brings security concepts with IAM but not compatible with COMMONAPI.

Reliability & efficiency:

- Communication pattern usage, event vs polling? Repeated signals?
- TCP vs UDP?

Stateless

- SOA is associated to the principle of stateless of the service, in the reality it is not that easy to strictly follow the rule for car system.

TAKE-AWAYS

- Adoption of SOA is not applicable to all area in automotive due to reliability opens.
- Not proven it can replace signal based in critical / realtime area.
- It is not deployable up to Applications in some ECUs (IVI Android)
- It has an impact on the internal organization and on SW integration with partners, left shifting integration steps on the iterative v-cycle.
- It better splits roles and responsibilities between Network and SW teams.
- It clearly derisks integration camps with partners, reduces the joint checks iterations.
- It does maximize SW code reuse across platforms with better decoupling.
- It does reduce time to market for SW features (connected car).

Q/A