DENSO

Crafting the Core

Rule Based Arbitration

DENSO CORPORATION

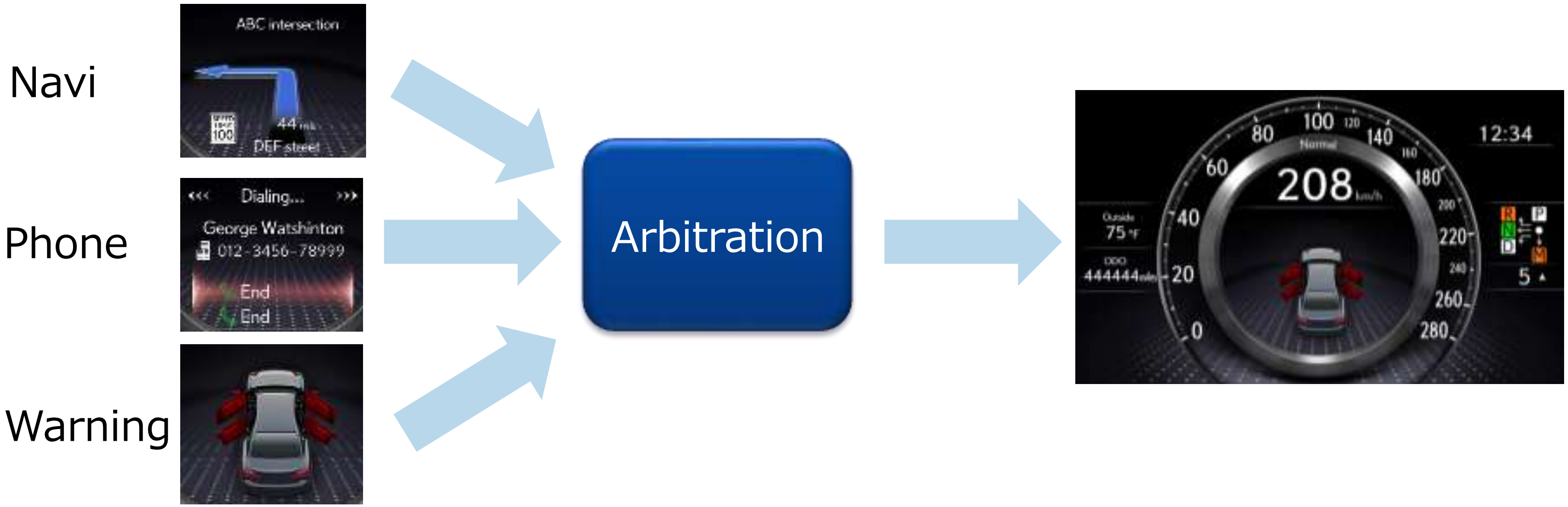
Advanced Driver Information Technology Corporation

Outline

- What is the Rule Base Arbitration
- Background
- The Advantage of Rule based arbitrator
- Rule Base Arbitration
    - Ex. Screen transition Spec.
    - What can be defined as a rule
    - Sample of basic rule definition
    - Sample of Exception rule def
- Software structure
    - Overview
    - Rule-based arbitrator structure
- Schedule

What is the Rule Base Arbitration?

When several information for driver (Content) needs to be notified simultaneously, RBA decides which content is prioritized.



Navi

Phone

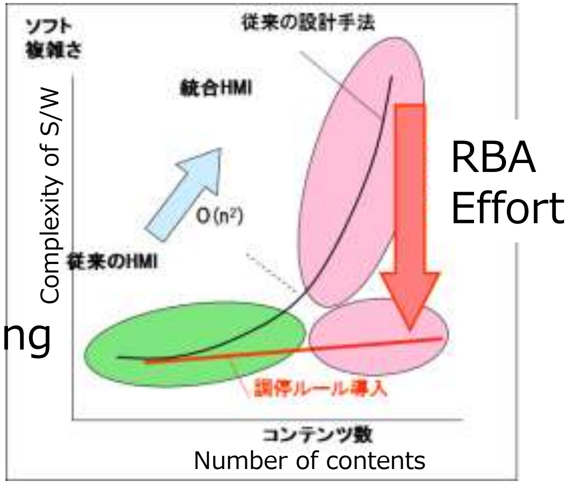
Warning

Arbitration

Background

- Issue of legacy technology:
  - Limit of status transition and Matrix
  - Contents are increased in every model.
  - Huge effort is needed for spec change.
  - Huge maintenance effort is needed due to existing spec is unclear.
- HMI Manager
  - Displaying preferable information to suitable area (display, position) based on driver’s character, preference, status and driving scene.
  - Flexible display arbitration for consolidated cockpit.

  ~Difficult to present by Status transition and Matrix~



Display arbitration will be more complex because of many scene, contents are increased for example autonomas driving.
Flexible arbitration logic is needed as base technology for realizing consolidation cockpit and HMI Manager concept

Rule Base Arbitration

-The Advantage of Rule based arbitrator

- **For OEM**
  - Intention/background of spec. can be ruled as it is.
    -> To prevent specs from becoming a dead letter
    -> To keep simple and high maintainability
  - Can confirm concrete behavior of spec. with simulator/actual hardware
  - Specification can be evaluated comprehensively.
- **For Tier-x Supplier**
  - To avoid complex software implementation.
  - Can reduce validation cost because spec has validated by OEM

New Point	Conventional	Rule based	Expected effect
Spec. def.	Manual creation	Automatic generation by tool	Production quality can be assured in early sample.
Rule def.	Filling arbitration rule matrix table	Constrains formula	
Product Software	Depends on HMI-FW	Independent of HMI-FW, OS	Reduce cost for developing

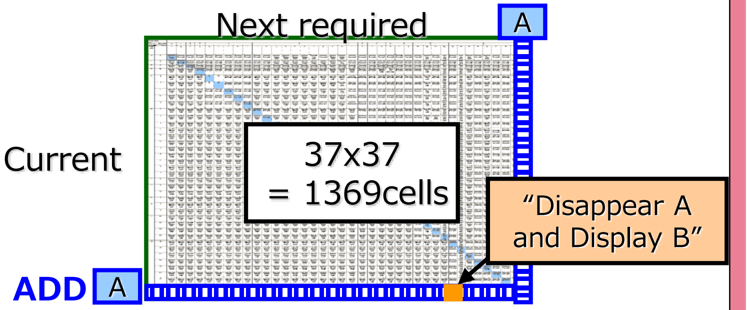
DENSO
Crafting the Core

Rule Base Arbitration

■ Legacy technology： Transition matrix

All behavior are defined in one matrix table.

Example: State transition design with table



Once A is added, all the relationships with other display contents should be considered

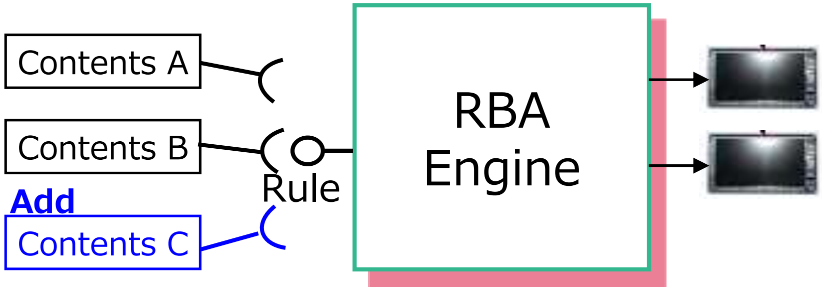
【Problem】

Many combination is increased for arbitration matrix, even if only one content is added.
-> Increasing much effort.

■New technology：Rule base design

Contents displaying policy are defined as abstracted rule and judge by RBA engine.

Example: Rule based design



Once C is added, only define the rule to apply to C

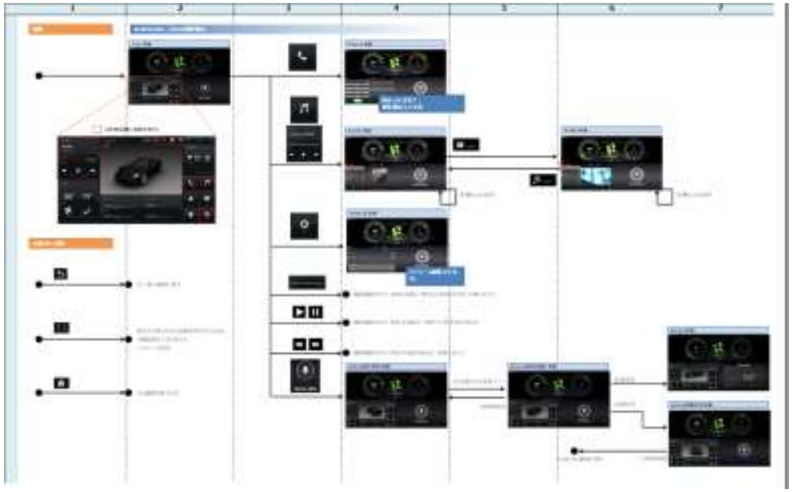
【Expected effort】

Even if new content is added, no affect to other content because RBA engine judges the display contents based on defined rule.
-> Saving effort

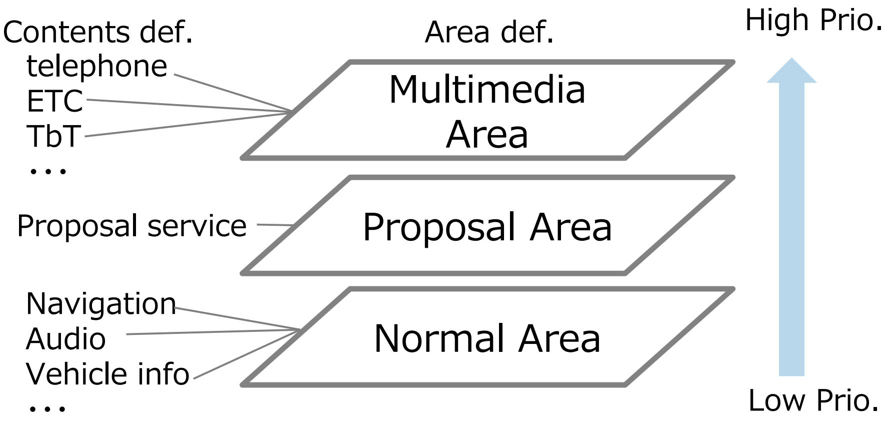
DENSO
Crafting the Core

Rule Base Arbitration - Ex. Screen transition Spec. -

Conventional : State machine



Rule based



Contents def.

telephone

ETC

TbT

・・・

Area def.

Multimedia Area

Proposal service

Proposal Area

Navigation

Audio

Vehicle info

・・・

Normal Area

High Prio.

Low Prio.

Basic Rule

・Higher priority wins between areas
・Later wins inside the area

Exceptional rules

・TbT notification is not displayed while navigation is displayed
・Low prio. contents is not displayed while telephone is displayed

- Difficult to add new content
- Difficult to understand intension or background of specification
- Difficult to define exceptional transition (such transition is described as remark)

- Easily add new contents
- Simple description
- Easy to understand background or reason of specification
- Behavior it not institutive.

Rule Base Arbitration - What can be defined as a rule -

- **Basic Rules**
  - Area definition(arbitration order, Z-order)
  - Arbitration policy
  - Content

    -> Priority, behavior of arbitration result(cancel, waiting)
  - Models for state transition (TAB screen transition in meter)
- **Exceptional Rules**
  - Constraint formula

    (Logical formula using status of are or contents)

    Logical operators: AND, OR, Implication, Compare, ∀、∃ and so on.
  - Exception behavior when losing in arbitration

    e.g. Cancel only when losing to specific content (usually waiting).

- **And more**
  - Arbitration of operation rights
  - Animation definition when transition

Rule Base Arbitration - Sample of basic rule def. -

Contents

```
ViewContent TEL {
                loserType: GOOD_LOSER
                allocatable: [MM_AREA]
                State OUTGOING {
                   priority: STANDARD_VALUE
                }
                State INCOMING {
                   priority: STANDARD_VALUE
                }
                State LIST {
                   priority: STANDARD_VALUE
                }
                sizeReference:Centralsize
}

ViewContent ETC {
                loserType: GOOD_LOSER
                allocatable: [MM_AREA]
                State NORMAL {
                   priority: STANDARD_VALUE
                }
                sizeReference:Centralsize
}
ViewContent VR {
                loserType: GOOD_LOSER
                allocatable: [MM_AREA]
                State NORMAL {
                   priority: STANDARD_VALUE
                }
                sizeReference:Centralsize
}

......
```

Layout

```
Package Displays {
      Display ICDISP {
      description:"IC"
      sizeReference: DisplaySize
      CompositeArea ICDISP_Root {
                  layout: FixedPositionLayout {
                        PositionContainer {
                                    x: 0
                                    y: 0
                                    basePoint: LEFT_TOP
                                    areaReference: BGarea
                        }
                        PositionContainer {
                                    x: 240
                                    y: 210
                                    basePoint: LEFT_TOP
                                    areaReference: MM_AREA
                        }
....
}
```

Area

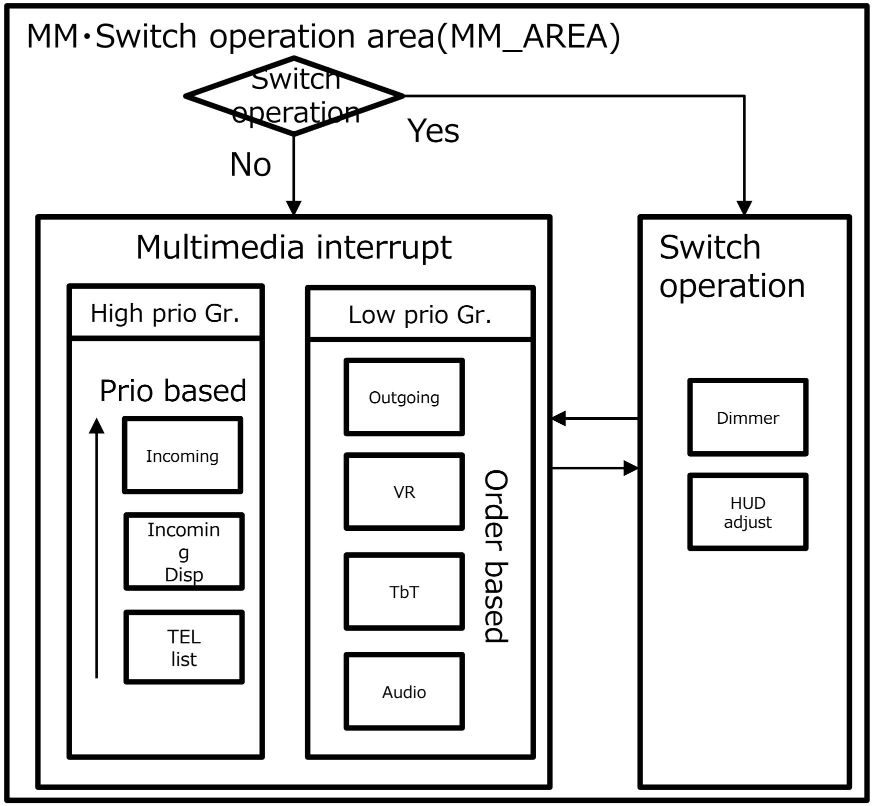
```
Area MM_AREA {
                  description:"MM_INTR"
                  arbitrationPolicy: LAST_COME_FIRST
                  sizeReference: Centralsize
                  visibility: > That-of Services･OprAdvisory
                  zorder: > That-of Services･OprAdvisory
}
Area VEHICLE_INTR {
                  arbitrationPolicy: PRIORITY_LAST_COME_FIRST
                  sizeReference: Centralsize
                  visibility: > That-of MM_AREA
                  zorder: > That-of MM_AREA
}
```

Syntax spec is being documented.

DENSO
Crafting the Core

Rule Base Arbitration - Sample of Exception rule def.

Screen transition spec



Conditions :

- Multimedia and Switch operation displayed on the same area
- Switch operation contents displayed by Switch operation
- Contents group with low and High prio defined in Multimedia interrupt area
- Contents group with low prio: New contents overwrites previous ones.
- Contents group with high prio: High prio contents overwrites low prio ones.

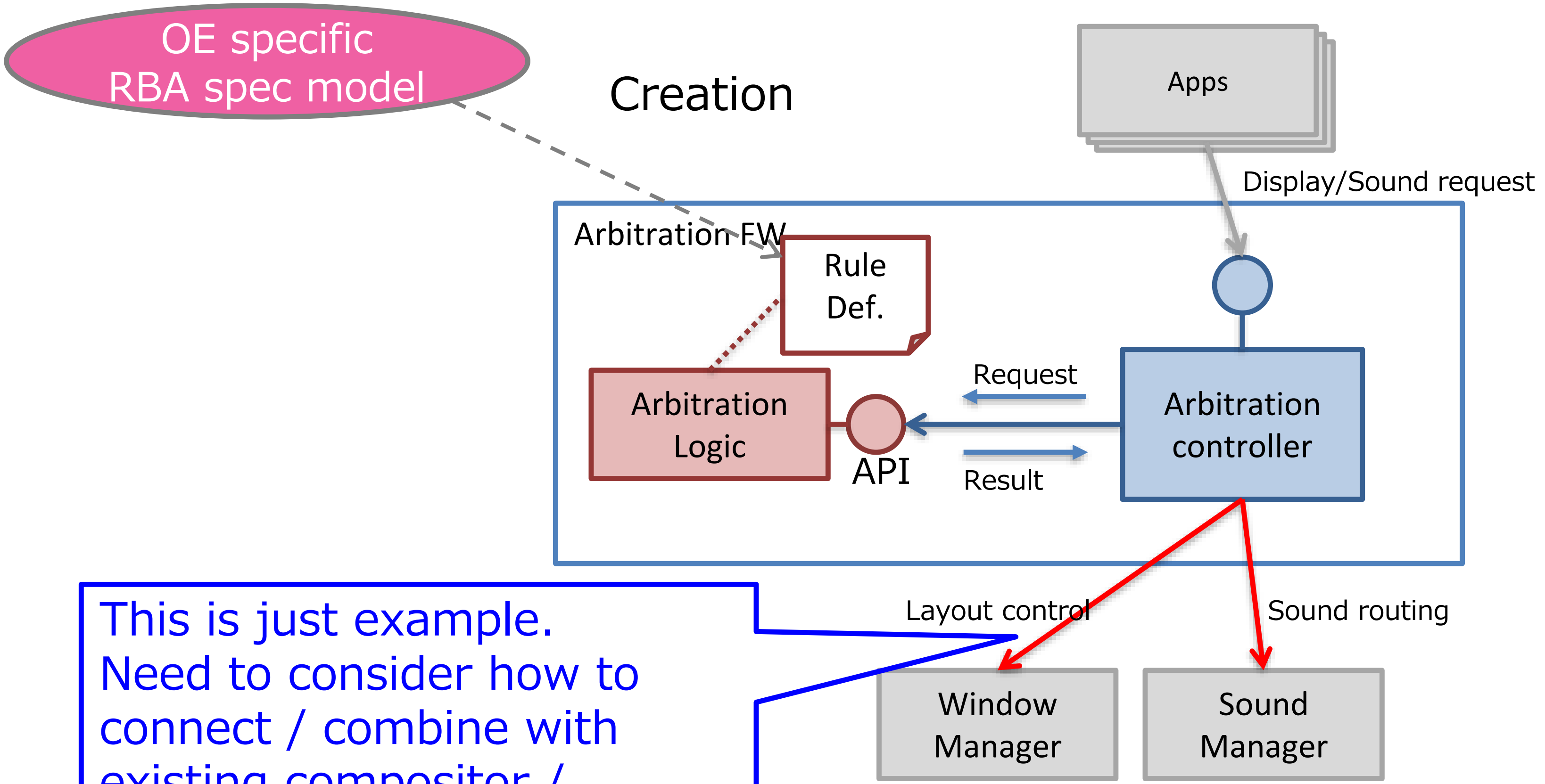
```
//MM_AREA: New contents basically overwrites old ones. But only Switch operation contents can be displayed during  TEL
contents displayed.

Constraint TEL with prio in MM_AREA {
            runtime: true
            (Exists MM_INTR_prioH { x | x.isActive() } AND  For-All SW_INTR { x | !x.isActive() })
              -> For-All MM_INTR_prioL { x | !x.isVisible() }
}
```

DENSO
Crafting the Core

Software structure – overview – 1/2

- **Option.1**

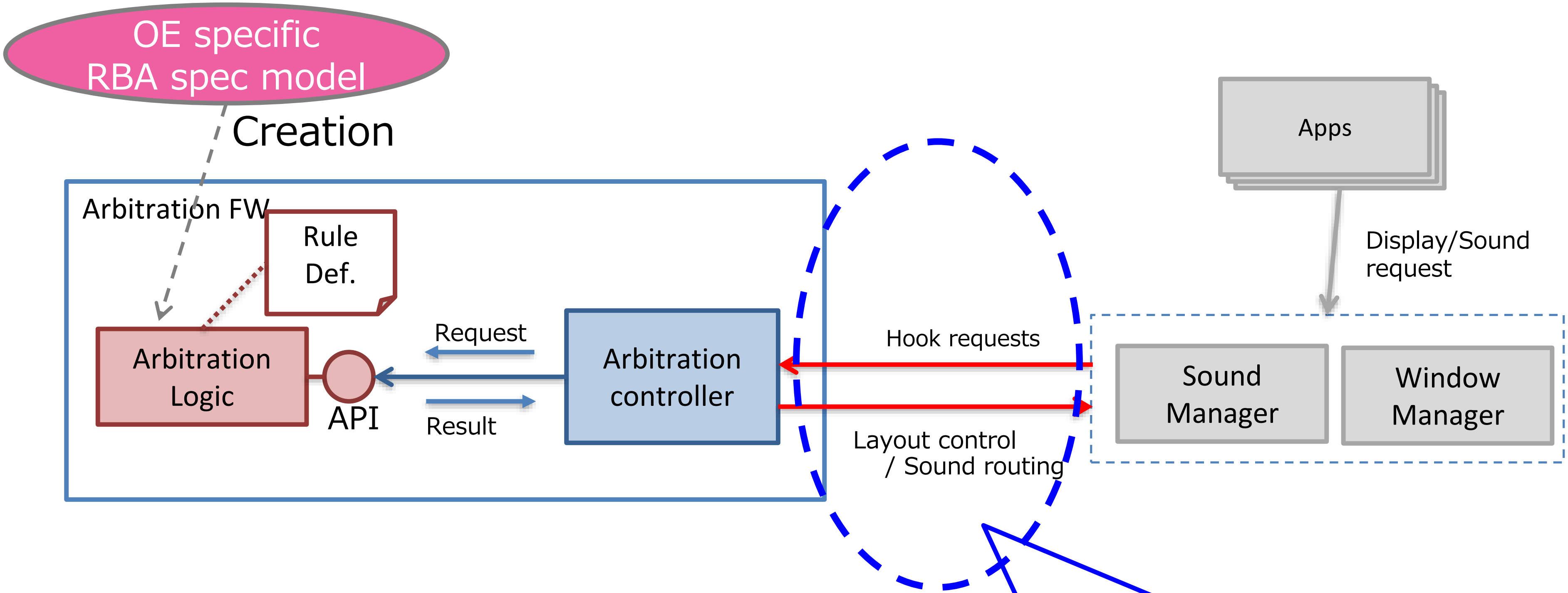


This is just example.
Need to consider how to connect / combine with existing compositor / wireplumeber.

RBA spec model makes it possible
to update result of arbitration, layout control and sound routing

Software structure – overview – 2/2

- **Option.2**

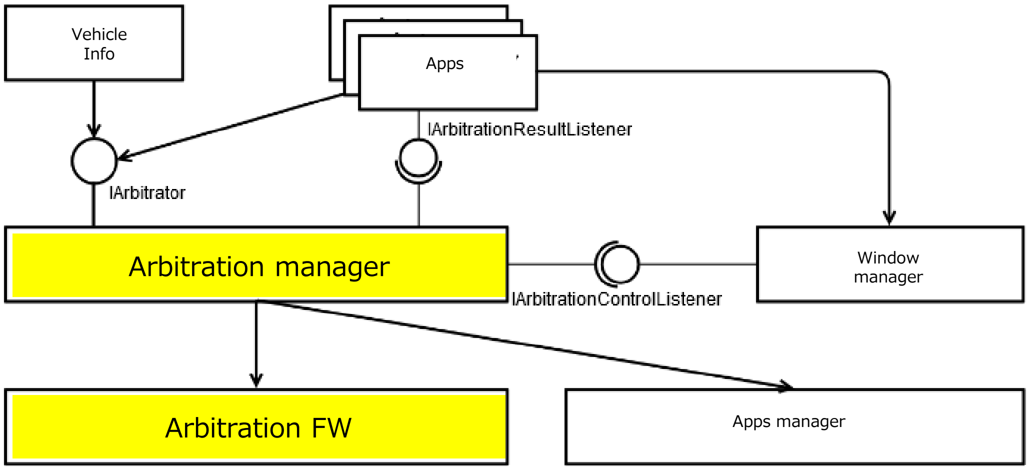


This is just example.
Need to consider how to connect/combine with existing compositor/wireplumeber.

Software structure - Rule-based arbitrator structure -

Basic func.

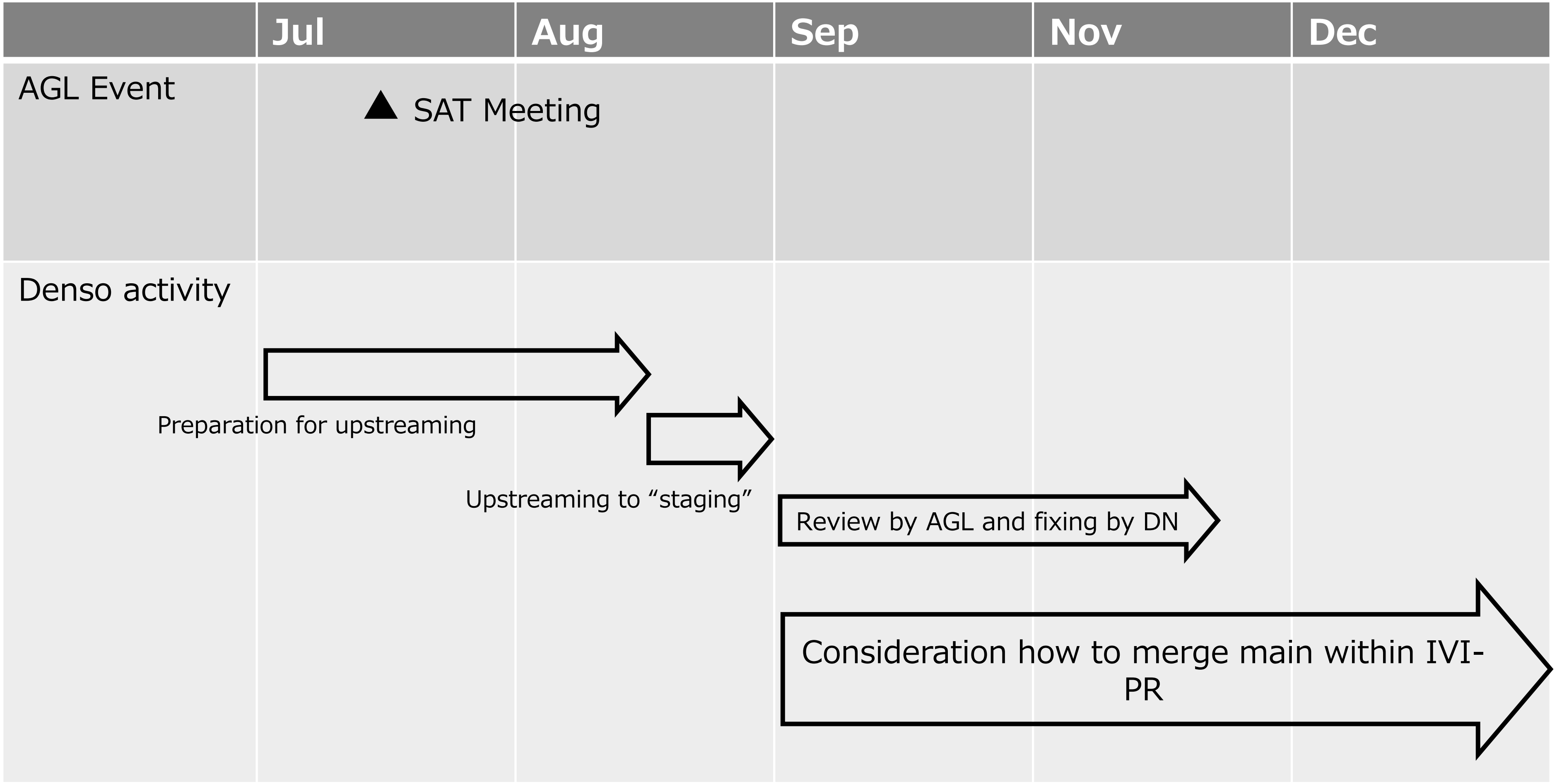
- Decide which contents shows at which area
- Arbitrate contents according to request from apps and scene (like power on/off, auto driving, ..)
- Notify arbitration result to apps
- The result contains difference from last result
- Synchronized multiple notifications bring no screen flickering



- **Arbitration manager** :
  - Receive contents request and scene info.
  - Arbitrate contents and notify the result to apps.
  - Notify start/end of arbitration to synchronize with Window manager.
- **Arbitration FW** :
  Arbitrate contents according to rule def.
- **IArbitrator Interface** :
  - Receive contents / scene request.
  - Manage registered apps
- **IArbitrationResultListener Interface** :
  - Receive arbitration result
- **IArbitrationControlListener Interface** :
  - Receive start/end of arbitration

DENSO
Crafting the Core

Schedule



Jul
Aug
Sep
Nov
Dec
AGL Event
▲ SAT Meeting
Denso activity
Preparation for upstreaming
Upstreaming to “staging”
Review by AGL and fixing by DN
Consideration how to merge main within IVI-PR

Currently, Upstream to AGL is in progress. -> https://github.com/NaohiroNISHIGUCHI/rba

Need to translate the Japanese comments in the code and header files.

DENSO
Crafting the Core

DENSO

Crafting the Core