DENSO
Crafting the Core

Rule Base Arbitration

Proposal of policy manager
from real product

DENSO CORPORATION
Advanced Driver Information Technology Corporation

Introduction

- Kenji Hosokawa
- HMI developer for IVI at Denso since 2005
- Graphics developer at ADIT since 2017
  - Wayland/Weston, Video input, GPU driver, DRM/KMS, ...

- Advanced Driver Information Technology
- Established in 2003
- Joint venture Denso and Bosch
- Produce IVI Platform for both MCs

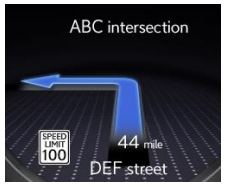
Outline

- What is the Rule Base Arbitration
- Background
- Rule Base Arbitration
  - Ex. Screen transition Spec.
  - What can be defined as a rule
  - The Advantage of Rule based arbitrator]
  - 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 at the same time, RBA decides which content is prioritized.

Navi



Phone

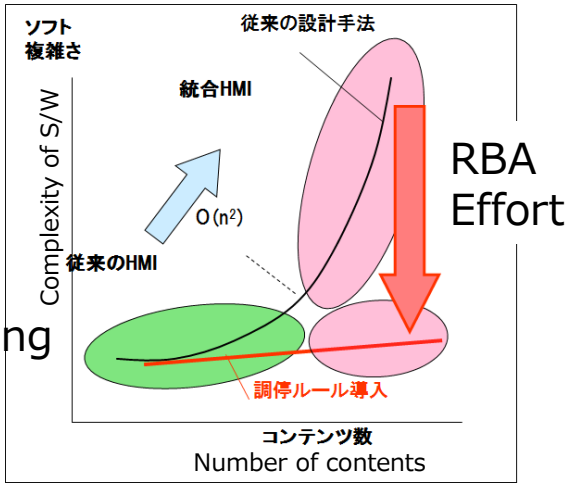
Warning

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~



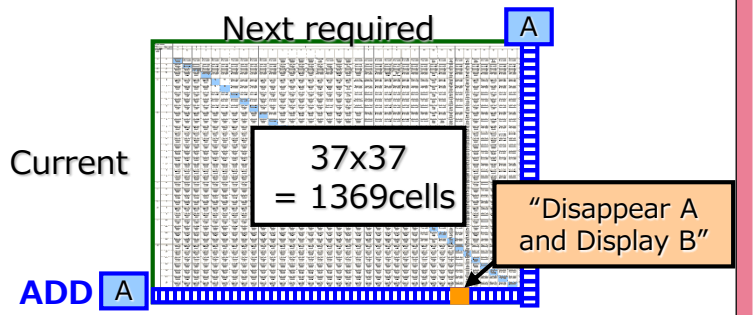
Flexible arbitration logic is needed as base technology for realizing consolidation cockpit and HMI Manager concept

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

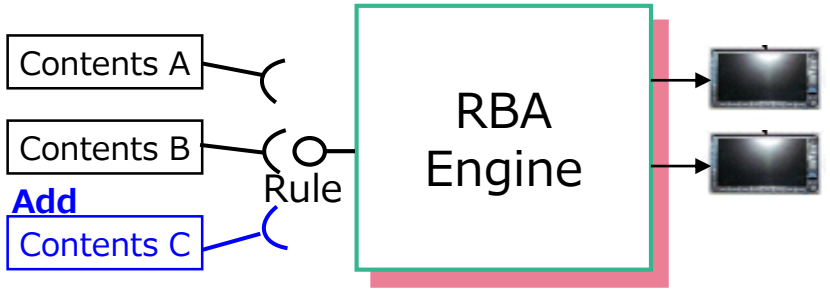
【Problem】

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



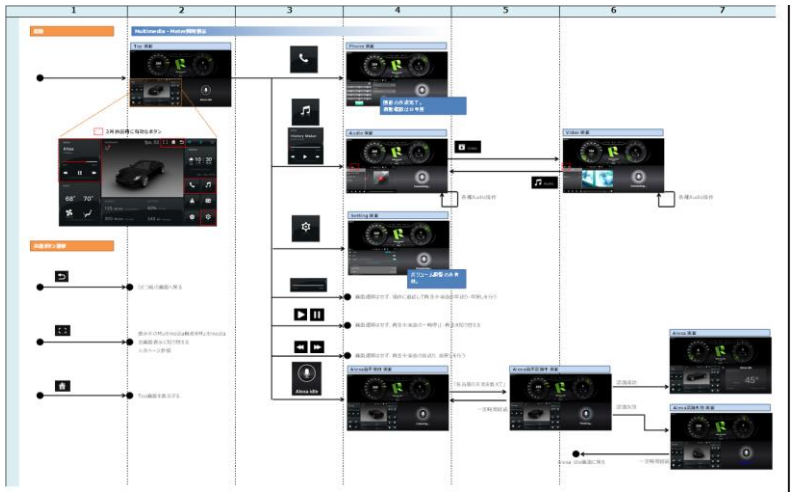
Once C is added, only define the rule to apply to C

【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

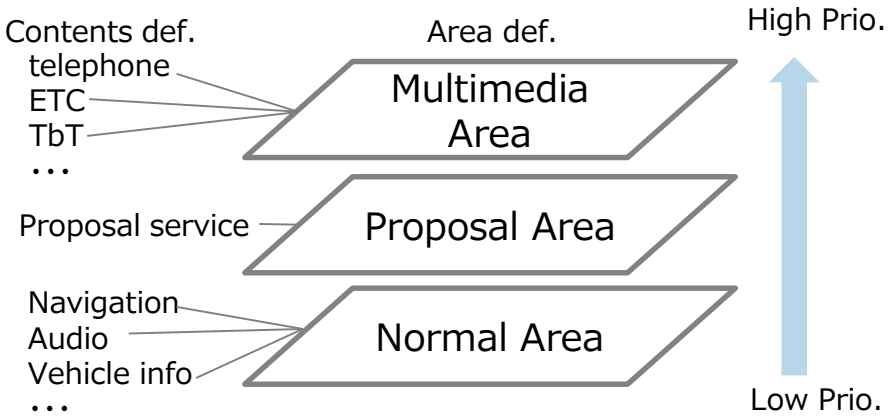
Rule Base Arbitration - Ex. Screen transition Spec. -

Conventional : State machine



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

Rule based



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

- Easily add new contents
- Simple description
- Easy to understand background or reason of specification

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

Please refer “Syntax definition”(which will be provided later) for more details

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 Supplier**
  - To avoid complex software implementation.
    Can reduce bugs by automatic code generation from spec.
  - 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	
Rule validation	Comprehensive manual testing	Automatic test by tool	Reduce cost for validation/test
Product Software	Depends on HMI-FW	Independent of HMI-FW, OS	Reduce cost for developing

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

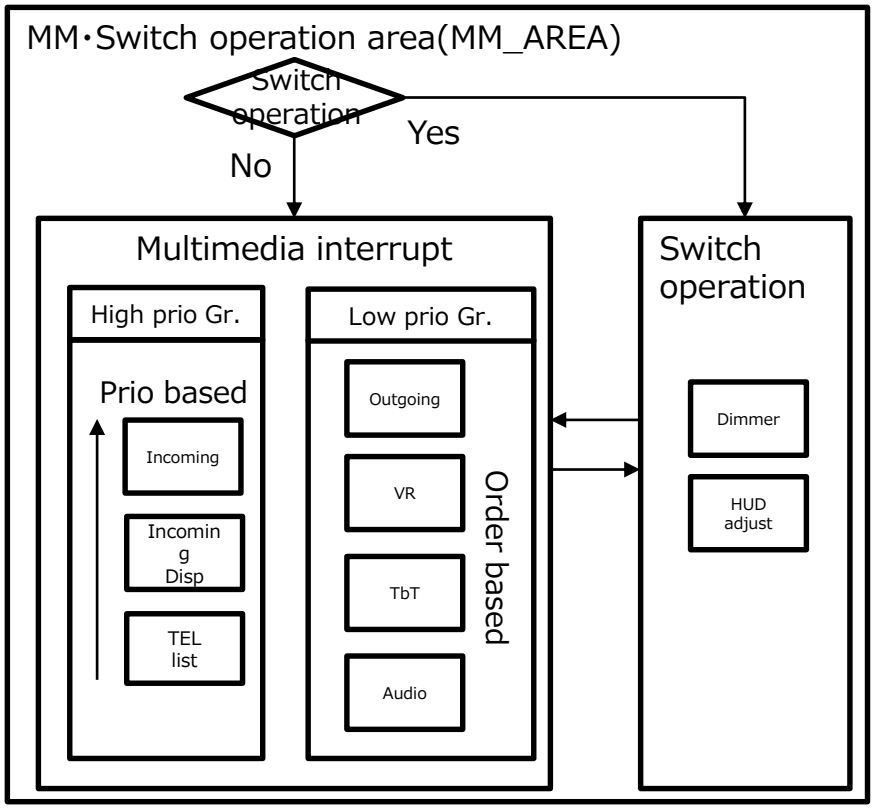
```
Package Displays {
    Display ICDISP {
    description:“IC"
    sizeReference: DisplaySize
    CompositeArea ICDISP_Root {
        layout: FixedPositionLayout {
            PositionContainer {
                x: 0
                y: 0
                basePoint: LEFT_TOP
                areaReference: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

Rule Base Arbitration - Sample of Exception rule def.

Screen transition spec



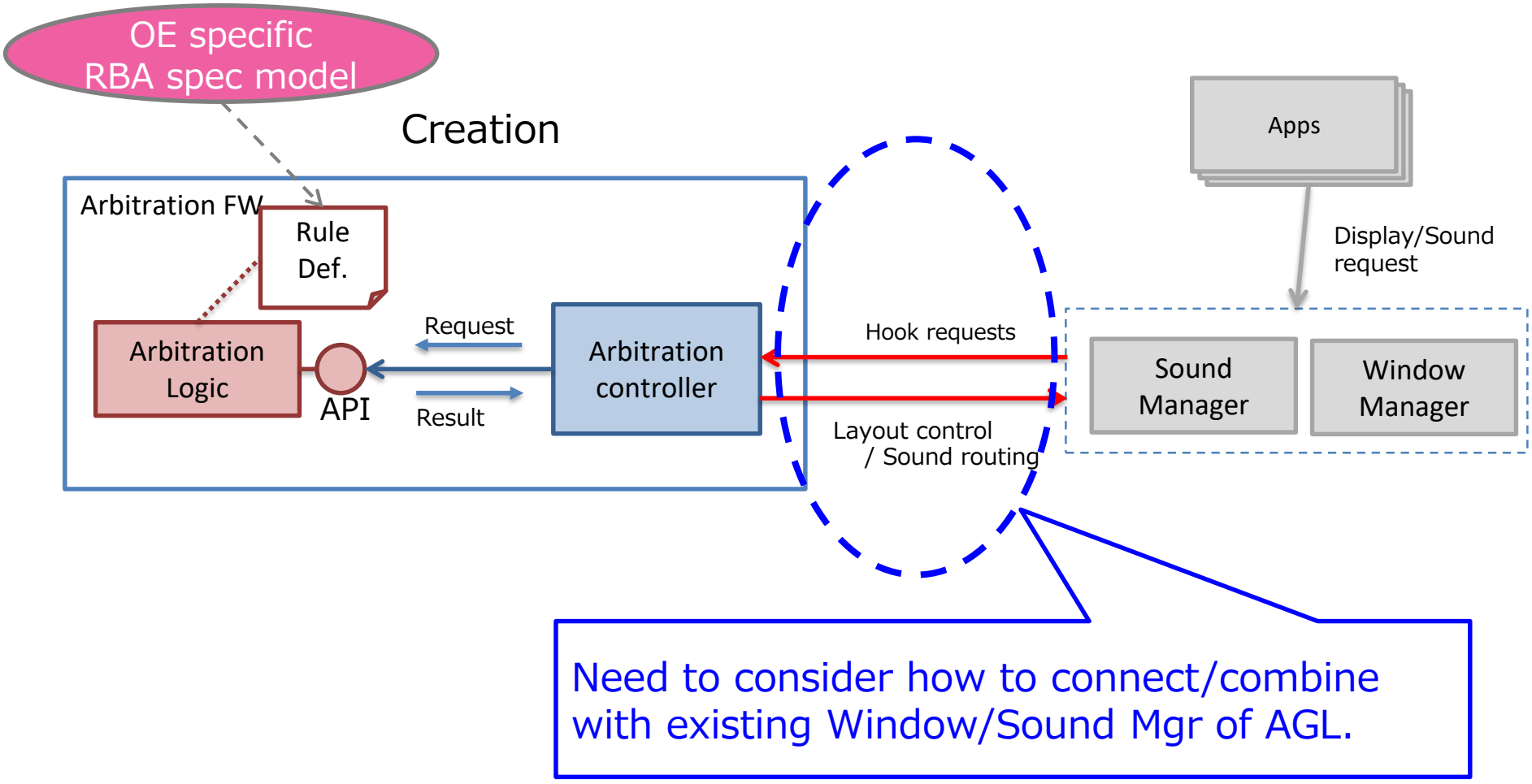
Conditions :

- 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() }
}
```

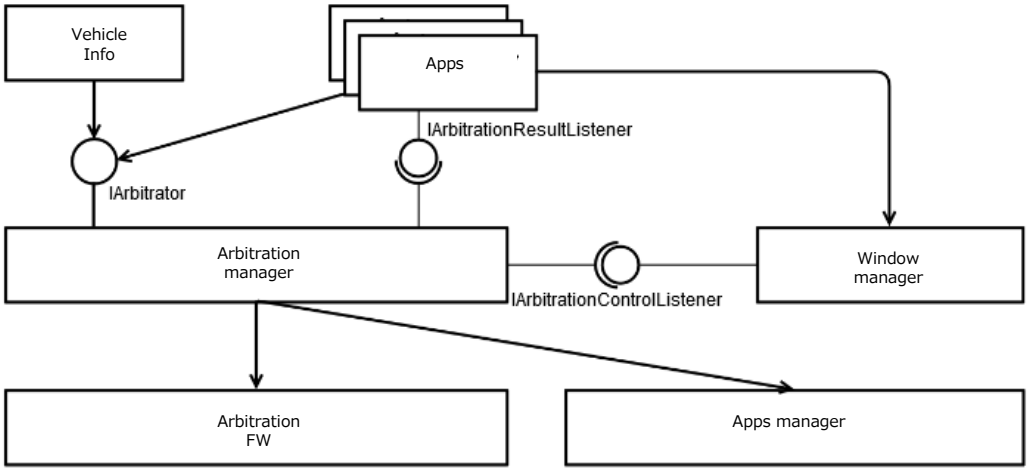
Software structure – overview –



Software structure - Rule-based arbitrator structure -

Basic func.

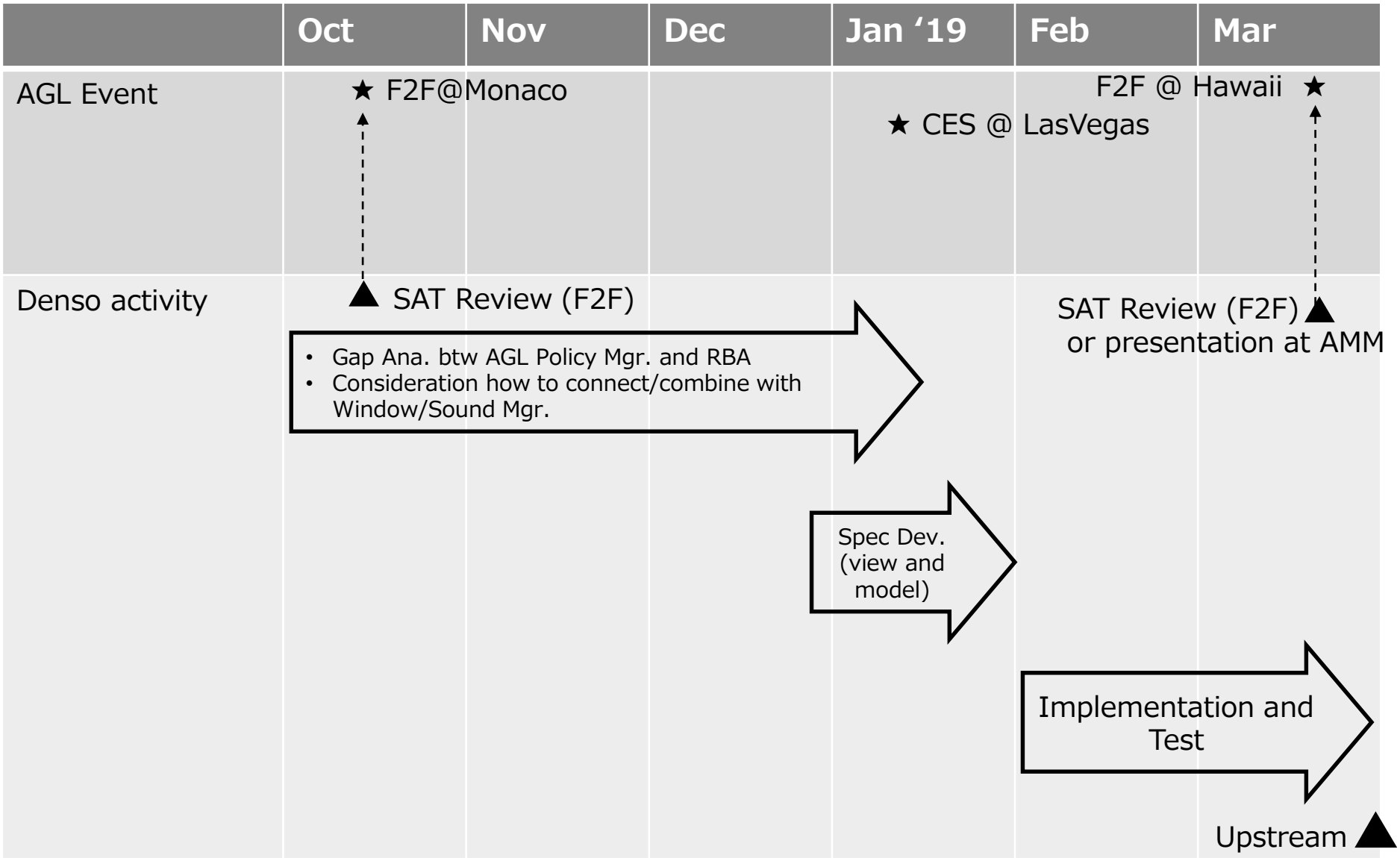
- Decide which contents shows at which area
- Arbitrate contents according to request from apps and scene like power on/off, auto driving, …)
- Notify arbitration result to apps
- The result contains difference from last result
- Synchronized multiple notifications bring no screen flickering



- 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 I/F :
  - Receive contents / scene request.
  - Manage registered apps

  IArbitrationResultListener I/F :
  - Receive arbitration result
- IArbitrationControlListener I/F :
  - Receive start/end of arbitration

Schedule



DENSO

Crafting the Core