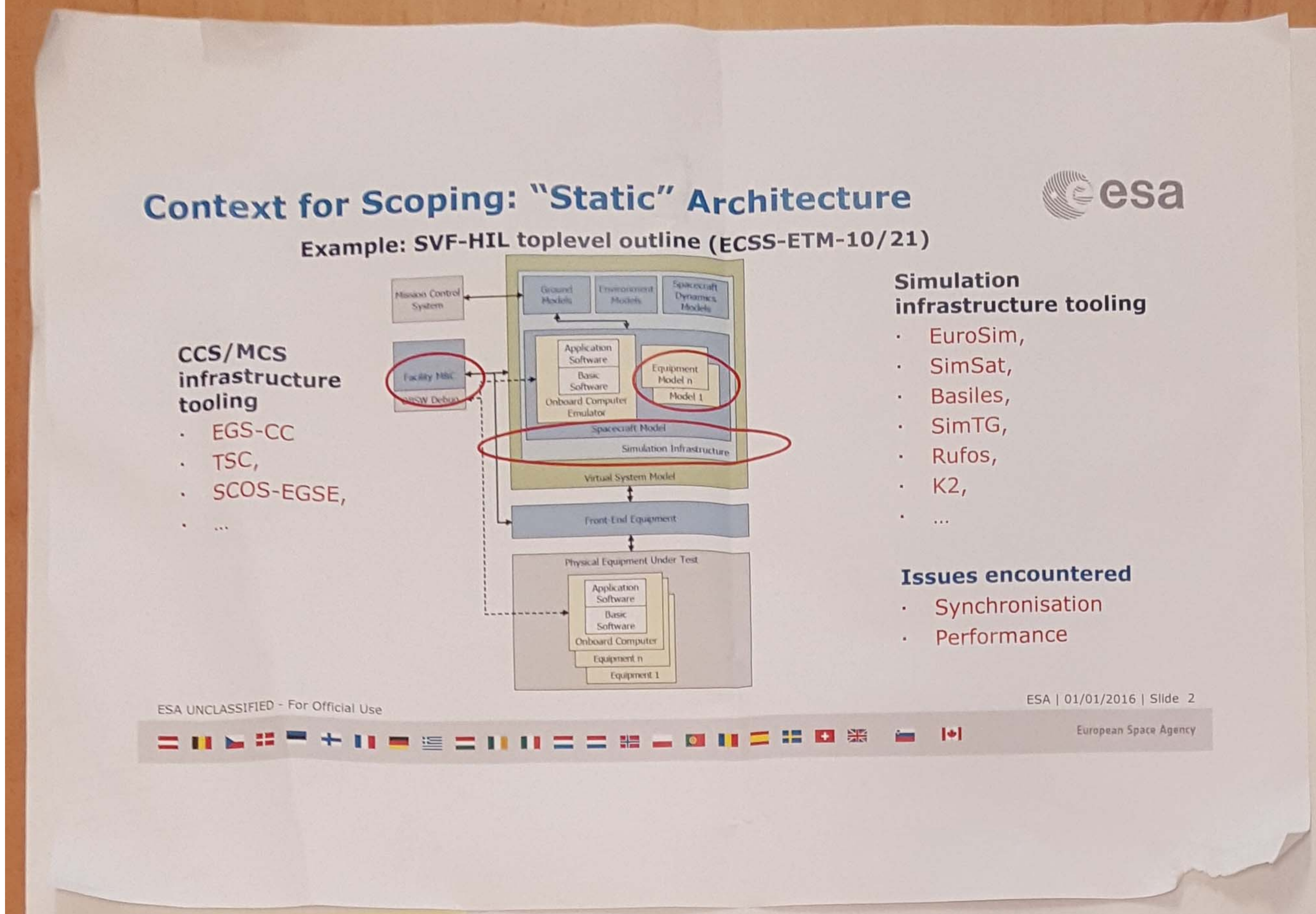
Panel 3: What to Rationalise?



Context for Scoping: "Static" Architecture

Example: SVF-HIL toplevel outline (ECSS-ETM-10/21)

CCS/MCS infrastructure tooling
- EGS-CC
- TSC,
- SCOS-EGSE,
- ...

Simulation infrastructure tooling
- EuroSim,
- SimSat,
- Basiles,
- SimTG,
- Rufos,
- K2,
- ...

Issues encountered
- Synchronisation
- Performance

ESA UNCLASSIFIED - For Official Use

ESA | 01/01/2016 | Slide 2

PLUG-IN
INTERFACES
(BB) around
Kernel

Rationalize Interfaces
and Processes
but not Tools

Interfaces to
simulate system
interfaces of equipment

Interfaces between
models made by different
vendors enforce standard
"ports"

BUILDING BLOCKS
COMPATIBLE AT
THE I/F
SHOULD SUPPORT
THE GEO RETURN

Simulat° Kernel first
↳ Interfaces to plug
custom/specific solut°
for specific needs
- needs to customize kernel

RATIONALIZE ARCHITECTURE
THIS ALLOWS DIFFERENT IMPLEMENTAT.
WITH INDIPENDENCE FROM TECNOLOGY
SMES AND NEW COMERS CAN ALWAYS
ENTER WITH INNOVATIVE OR
"BETTER PERFORMANCE" SOLUTIONS OVER
THE TIME

Established Open Source
solutions, which may
require adaptations
for space industry

HIL TOOLING
- Drivers
- plug ins
- HW.

if Tool
THAN

Reusable SCOE
across mission.

· Include:
Mission Performance Sims.
⇒ Same tools needed
⇒ Different content.

Follow-up to
SSRA needed

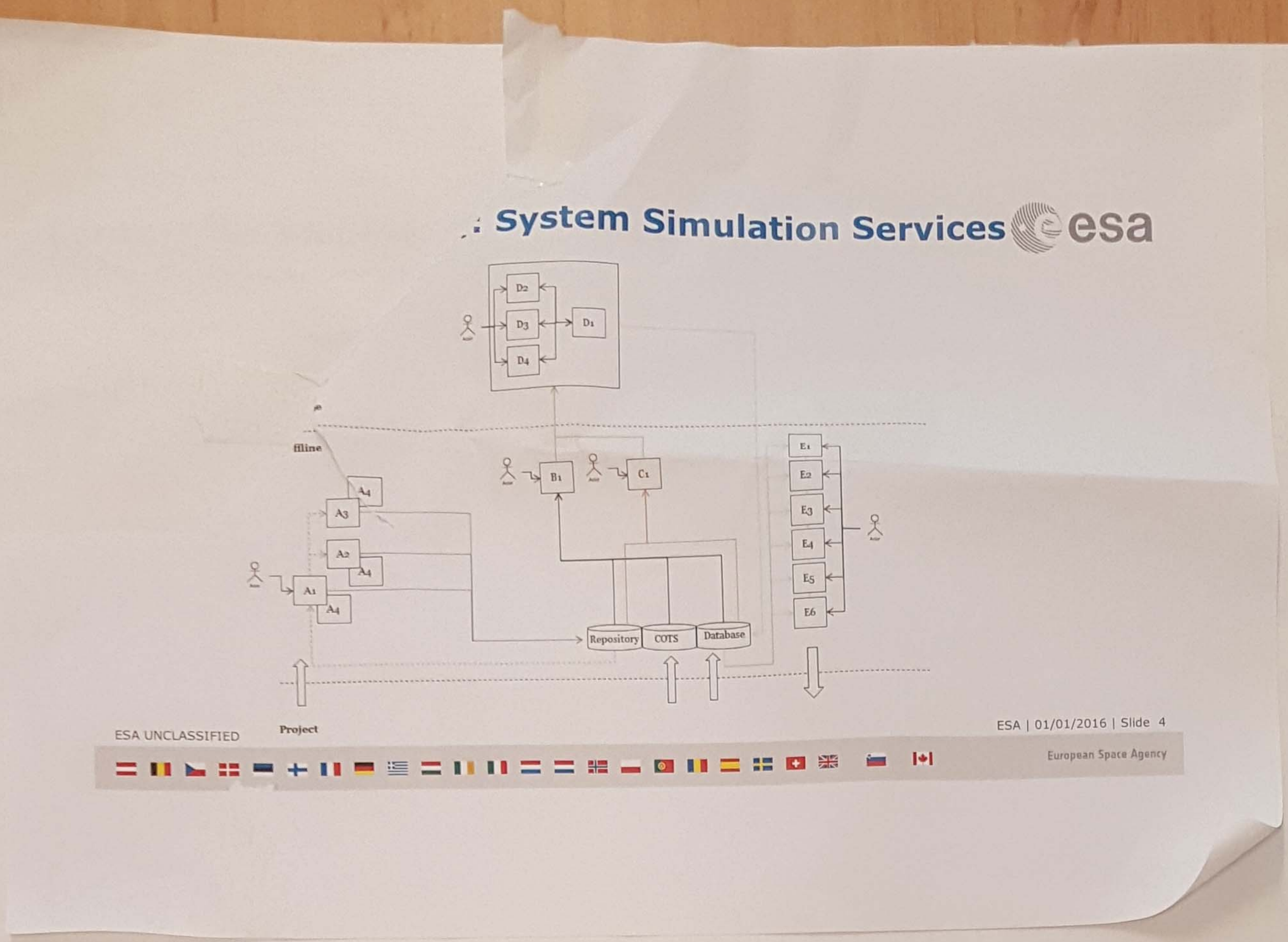
- BUILDING
BLOCKS
Examples: AOCS HW/
equipment simulation
models/modular FES → Avionics [illegible]

Items for Rationalisation
* New technologies/features
that are on the ToDo-
List of all stakeholders
* existing intellectual props.
are not affected

Enable exchange of
models not tools
⇒ promote SMP2 L1&L2
compatibility

Reference Architecture taking
into account dynamic models:
thermal, attitude, orbit,
ephemeris, ...

- A Core common to
all different applications
which is rationalised
- A set of specific plugins
and libraries
- The rest to be free



: System Simulation Services

ESA UNCLASSIFIED

ESA | 01/01/2016 | Slide 4

European Space Agency

- need to "right size"
i.e. simple users do not want to
have to install a heavy infrastructure

1. Reference Architectures
2. Design Tools / Tools (SMP2)
3. Processes

BUILD FLEXIBLE AND
OPEN, AND MODULAR, ARCHITECTURES
BASED ON SOFTWARE
DESIGNED INSTRUMENTS

TESTING METHODO-
LOGIES
- TEST CASES
- SCENARIOS
-

1) Standards
2) Referene Architecture
3) Implementation (optional)

HLA, DDA, ... DOESN'T MATTER
MATWORKS (MATLAB & SIMULINK)
WILL OFFER FULL MODULARITY.
THIS IS OUR COMITMENT
-MATHWORKS-

BEFORE HAVING
GENERIC BB, WE
SHOULD CONVERGE ON
GENERIC FUNCTIONAL
SPECS (ALL LEVELS)
(e.g Specs for SVF || Gyro)

RATIONALISATION STEPS
1. PROCESSES/STANDARDS
2. REFERENCE ARCHITECTURES
(ONE PER TARGET SIMULATOR
TYPE)
3. DEVELOP/INTEG/EXECUTION TOOLS

1) Reference Arch.
2) Configuration
3) tool chain (MMI, Post Proc.)

- Reusable SCOEs across missions
- Include: Mission Performance Sims-> same tools needed -> different content
- **Follow-up to SSRA needed**
- Plug-in interfaces (BB) around kernel
- Rationalize architecture: this allows different implementations with independence from technology SMEs and new comers can always enter with innovative or "better performance" solutions over the time
- Building blocks, examples: AOCS HW/equipment simulation models (modular FES-> Avionic Bench)
- Rationalize interface and processes but not tools
- Established open Source solutions, which may require adaptations for space industry
- Items for rationalization:
    - New technologies/features that are on the ToDo list of all stakeholders
    - (make sure that) Existing intellectual properties are not affected
- Rationalize: **Interfaces to simulate system interfaces (of equipment)**
- What to standardise: HIL tooling (if we are to rationalize tooling than):
    - Drivers
    - Plug-ins
    - HW
- Enable exchange of models not tools-> **promote SMP2 L1 & L2 compatibility**
- Mind the Right Size: e.g. Simple Users do not want to have to install a heave infrastructure
- Rationalize:
    - **Reference Architecture**
    - Design Tools (SMP2)
    - Processes
- Build Flexible and Open and Modular Architectures based on Software Designed Instruments
- Testing Methodologies
    - Test Cases
    - Scenarios
- Rationalize:
    - Standards
    - **Reference Architecture**
    - Implementation (optional)
- Rationalisation Steps:
    - Processes/standards
    - **Reference Architectures**
        - One per target simulator type
    - Develop/Integrate/execution tools
- HLA, DDA, … does not matter:
    - Mathworks (Matlab/Simulink will offer full modularity)
    - This is our commitment -- Mathworks --
- Before having Generic BB, we should converge on Generic Functional Specifications (at all levels)
    - (e.g. at level of SVF but also at level of Gyro)
- What to Rationalize:
    - **Reference Architecture**
    - Configuration
    - Tool Chain (MMI, Post Processing)