a partitioned pok operating system

POK - Introduction

Julien Delange <julien@gunnm.org>

Forewords

a partitioned pok operating system

- **The POK project**
  - **Design and implement safe and secure system**
  - **Complete development process with model-based engineering**
- **POK O/S**
  - **Part of the POK project**
  - **Can be used independently**
  - **Support several standards (ARINC653/POSIX/...)**

Outline

a partitioned pok operating system

- POK rationale and overview
- POK objectives and guidelines
- Functionalities
- Platforms

Outline

a partitioned pok operating system

- **POK rationale and overview**
- **POK objectives and guidelines**
- **Functionalities**
- **Platforms**

Rationale - problems

a partitioned pok operating system

- **Safety-critical systems are difficult to build**
  - **Many requirements**
  - **Dedicated architectures**
  - **Specific standards: ARINC653, MILS, DO178B, ECSS, etc**
- **Need to validate/verify/specify/certify**
  - **At specification**
  - **At execution**
  - **Require an appropriate development process**

Rationale – solution overview

a partitioned pok operating system

- **Framework to build safety-critical systems**
  - **From specifications down to the implementation**
  - **Enforce requirements at each development step**
- **Help designers as more as possible**
  - **But do not force him !**
  - **POK keeps you free !**

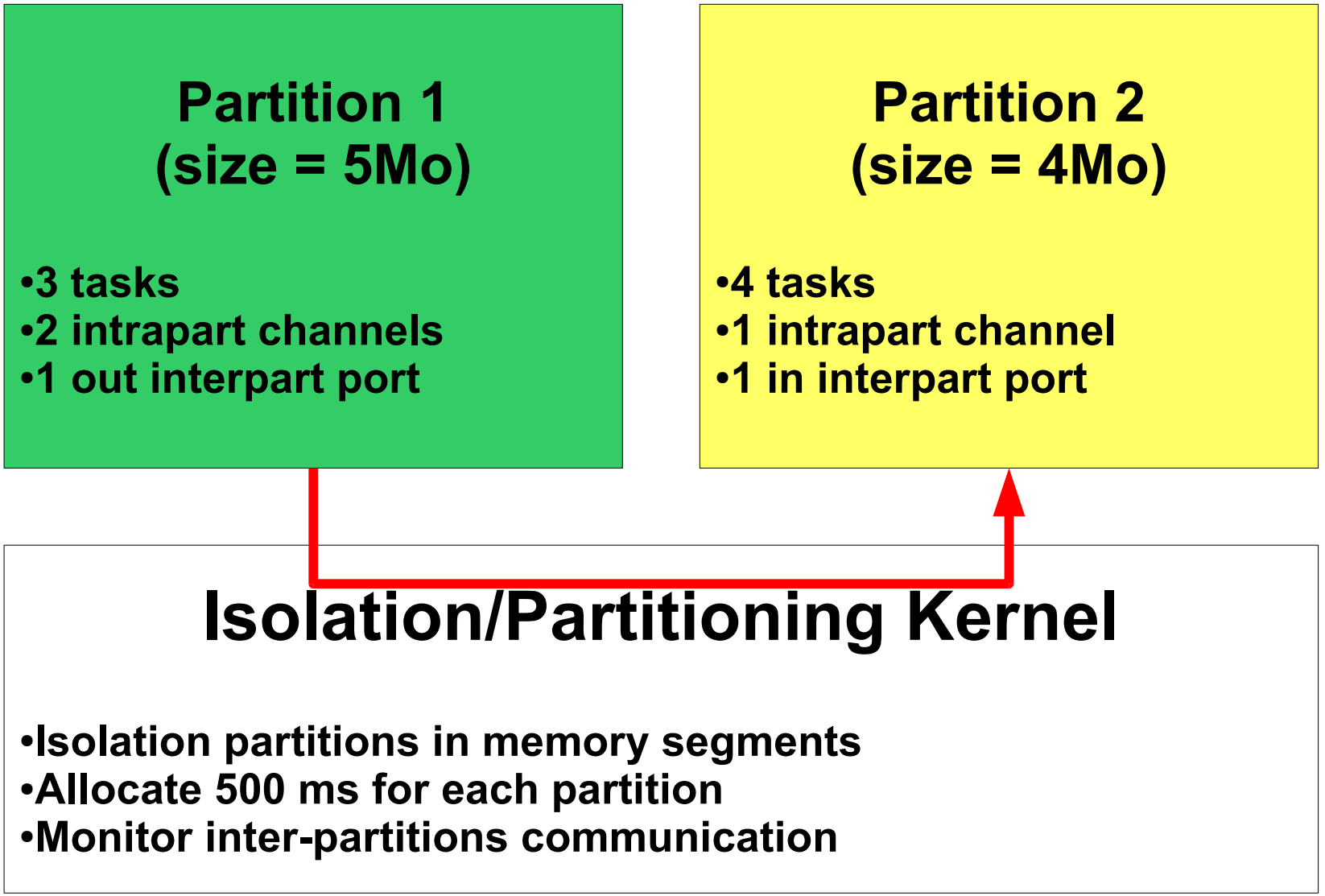
Solution – Architecture outline

a partitioned pok operating system

- **Partitioned architecture**
  - **Time and space isolation**
  - **Resources isolation across partitions**
- **Fixed amount of resources**
  - **Avoid run-time overhead**
  - **Improve determinism**

Solution – Architecture outline – cont'd

a partitioned pok operating system

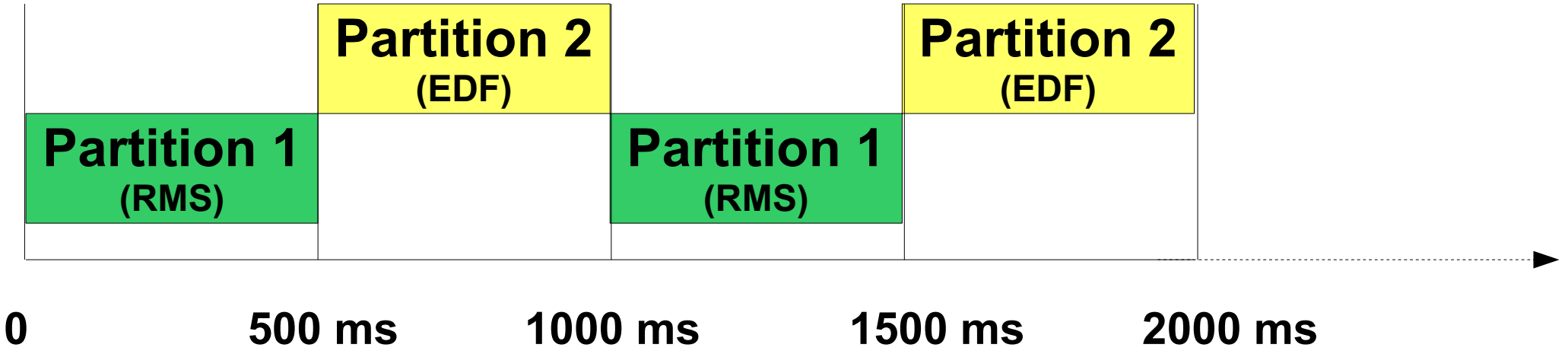


Architecture outline – space partiitoning

Organization of memory

- **Segmentation isolation**
  - **Determinism**
  - **Sometimes, emulate segmentation (e.g. LEON port)**
- **Allocate segments at initialization time**

Architecture outline – time partitioning



- Strict time isolation
- Round-Robin policy for partition scheduling
- Partition-dependent scheduling algorithm

Technology overview (1)

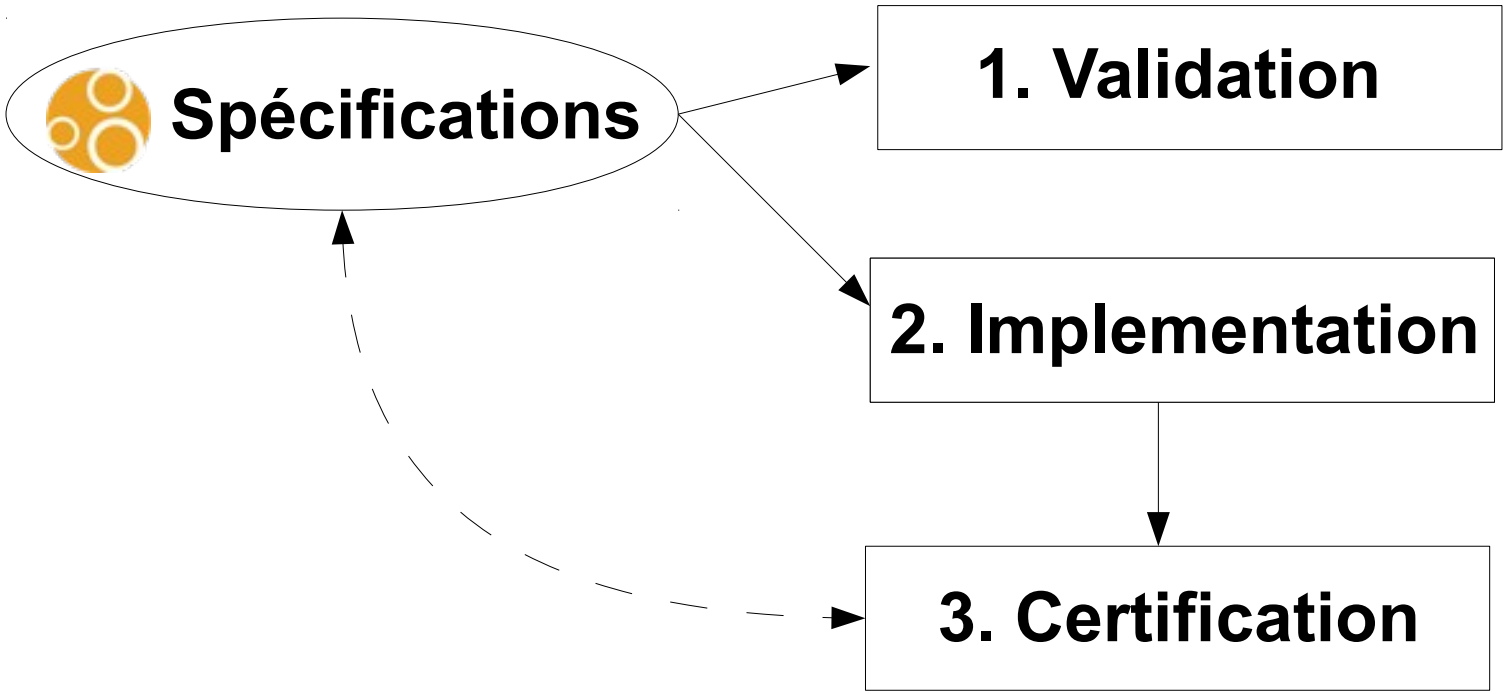
a partitioned pok operating system

- **Modeling framework (AADL modeling)**
  - **AADL modeling patterns**
  - **Requirements validation through AADL analysis**
- **Partitioned operating system & code generation**
  - **Provide time and space partitioning**
  - **Ensure requirements enforcement**
- **Certification tools**
  - **Verify implementation against specification requirements**
  - **Certify implementation against certification standards**

Technology overview – cont'd

a partitioned pok operating system

- **Full development process**
  - **Some steps could be achieved manually**
  - **We don't force developers to do good things**



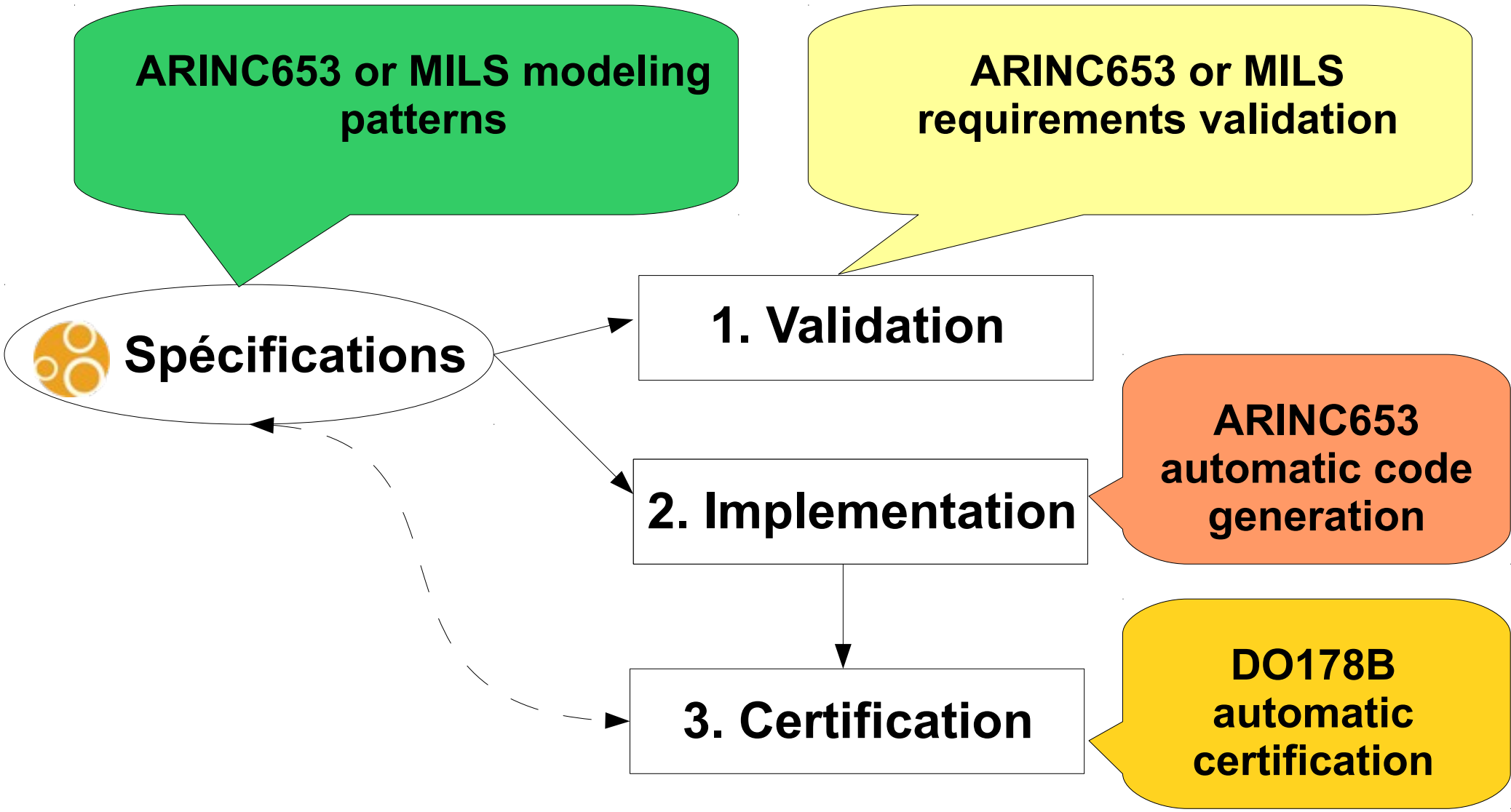
Relation with current work

a partitioned pok operating system

- **We propose a nice development process**
  - **Requirements validation and enforcement**
  - **Many tools that may be well developed by computer geeks**

- **But we want industrial solutions !**
  - **Standard compliance**
  - **Ex : ARINC653, MILS, etc ...**

Standards integration



Outline

a partitioned pok operating system

- POK rationale and overview
- **POK objectives and guidelines**
- Functionalities
- Platforms

POK objectives

a partitioned pok operating system

- **Build safe and secure systems**
  - **Provide a complete framework for the whole dev process**
  - **Detect errors as early as possible**
  - **Assist the user in each development step**

- **Potential industrial impact**
  - **Integrate industry-proven approaches**
  - **Development process can be adapted to customer needs**

POK guidelines

- KISS !
  - Very simple: must be easy to read/understand/ maintain ...
  - … but very, very stupid
- Keep the window open !
  - Potential kernel certification
  - Openness to other projects (ex : COUVERTURE)
- Highly-customizable toolchain
  - You can throw away some tools
  - Integration of other tools (ex : OSATE, Eclipse, ...)

POK functionnalities

a partitioned pok operating system

- **Specification-level**
  - **System consistency, safety & security validation**
  - **Automatic code generation for kernel & partitions**
- **Runtime level (POK O/S functionnalities)**
  - **Partitioning support**
  - **Standards supports : ARINC653, POSIX**
- **Certification level**
  - **Code coverage analysis of both kernel and partitions**
  - **Execution instrumentation**

POK architecture support

- X86 intel
  - Generic architecture
- SPARC
  - LEON3 board
  - Aerospace domain
- PowerPC
  - Wide-used architecture
  - Embedded-purpose domain

a partitioned pok operating system

Questions ?