SystemC-AMS as Key Component to DLR Spacecraft Power System Design Automation

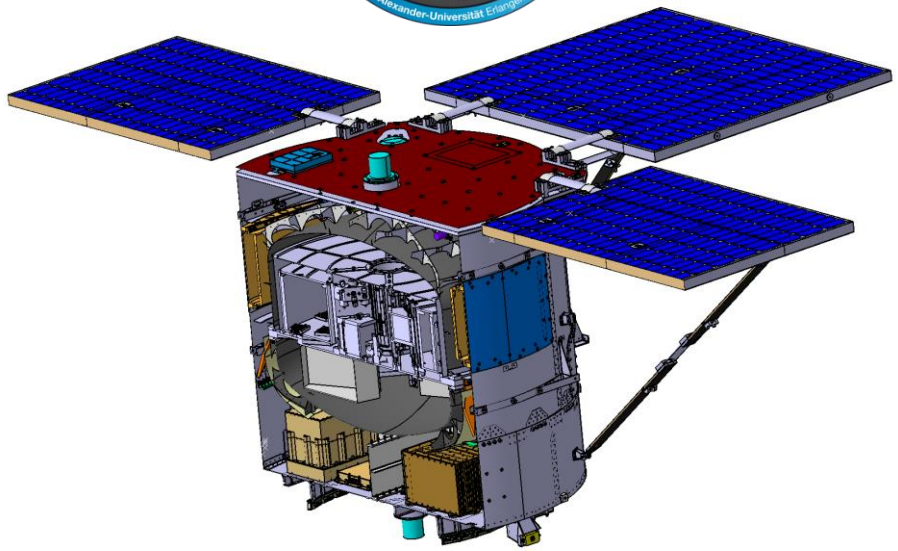
Janis Sebastian Häseker
German Aerospace Center (DLR) – Institute of Space Systems
Contact: janis.haeseker@dlr.de

Problems in Science Spacecraft System Design

- Heavily changing spacecraft bus requirements from mission to mission
- Budgetary and schedule constraints
- Latest science → low payload TRL* → bus requirements change late

This contradicts classical spacecraft design methodologies:

- Heavily based on heritage
- Reuse and manual adaptation of existing units
- Multiple (Hardware) models being tested extensively



Eu:CROPIS launched in 2018

*TRL: Technology Readiness Level

How are design processes accelerated in other industries?

- Automotive and Cyber Physical Systems (CPS)
  - Virtual Integration Testing
  - Rapid Prototyping
  - Correct by Design
- Integrated Circuit Design
  - Design Automation
  - Automated Design Validation/Verification
- Software Engineering
  - Libraries
  - Continuous Integration
  - Test driven development

What we want to adapt: Automate Design! Fail early! Continuous Design Validation/Verification!

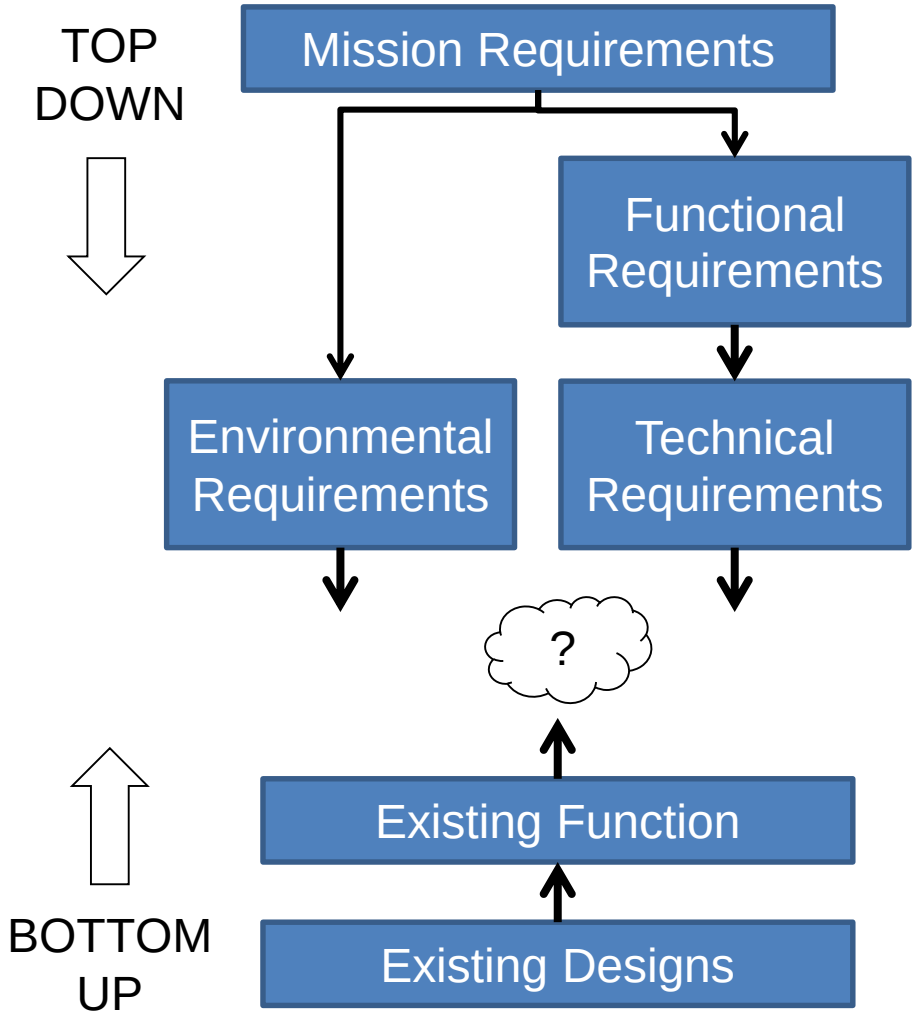
System Level Design and the Platform Based Design Methodology

- System Level Design
  - Model from highest level of abstraction
  - TOP DOWN to implementation

- Platform Based Design
  - „Meet in the middle“
  - Define Functionality TOP DOWN
  - Match Implementation BOTTOM UP

  → This methodology fits well for design automation
    - Find optimal design from library of implementations
    - Widen design space by adding hypothetical designs

  → Same modeling environment
    for TOP DOWN and BOTTOM UP



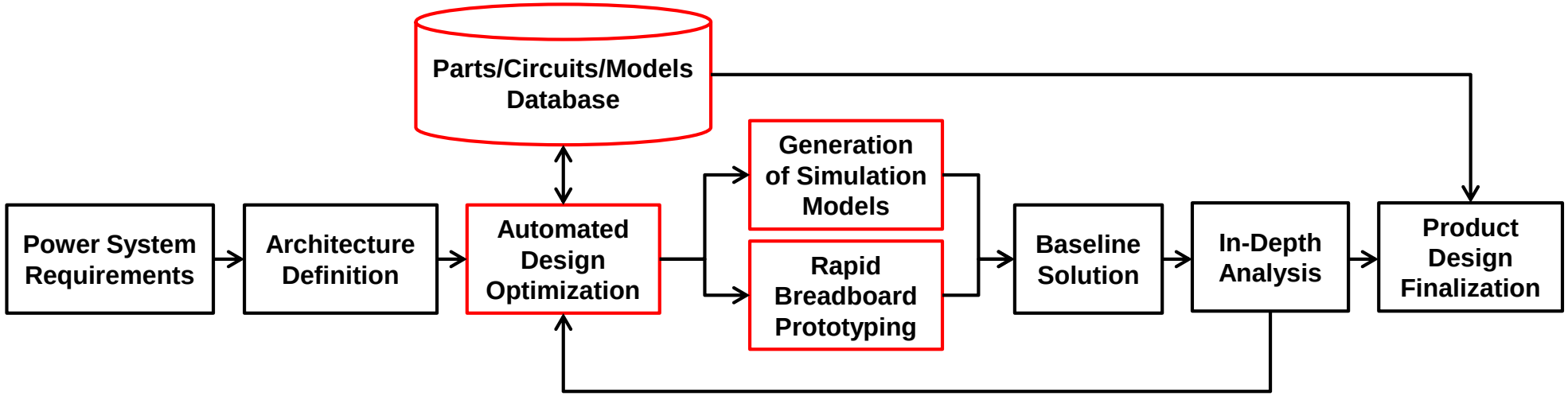
Modeling Language for Platform Based Design: SystemC-AMS

Requirements:

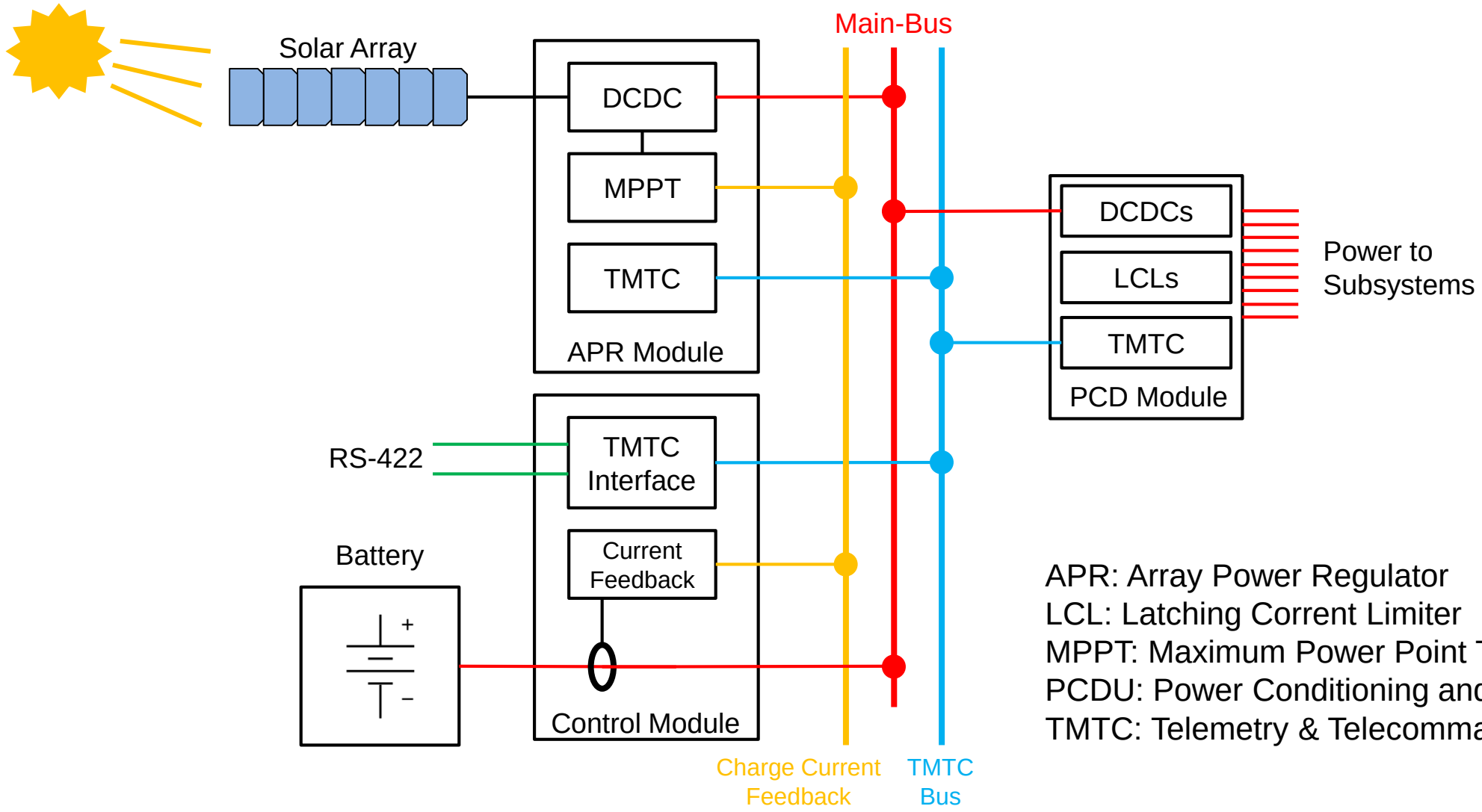
- Can be used for all relevant levels of abstraction
- Domain independent
- For hardware, software and environment
- C/C++ code and libraries can be used

Usecase: Scalable Power System Design

- Design Automation for Power Conditioning and Distribution Unit (PCDU) Development
- Correct by Design and Bespoke PCDU Solution for Scientific Nano/Micro/Mini Satellite Missions



Scalable Power System



APR: Array Power Regulator
LCL: Latching Corrent Limiter
MPPT: Maximum Power Point Tracker
PCDU: Power Conditioning and Distribution
TMTC: Telemetry & Telecommand

Design Automation Toolchain

- High Level Requirements and Spacecraft Architecture Capture: SysML
- Preliminary Power System Sizing Tools in Python & Excel
- Design Automation Scripting in Python
  - Architecture Definition
  - Design Optimization
  - Simulation Model Assembly
  - Analysis of Results
- Modeling of Hardware, Logic/Software and Environment in SystemC-AMS
- Continuous Integration Environment: GitLab

Goal: Lower the Level of Modularity

Classical S/C Design: Reuse as is or manual adaptation of existing designs.

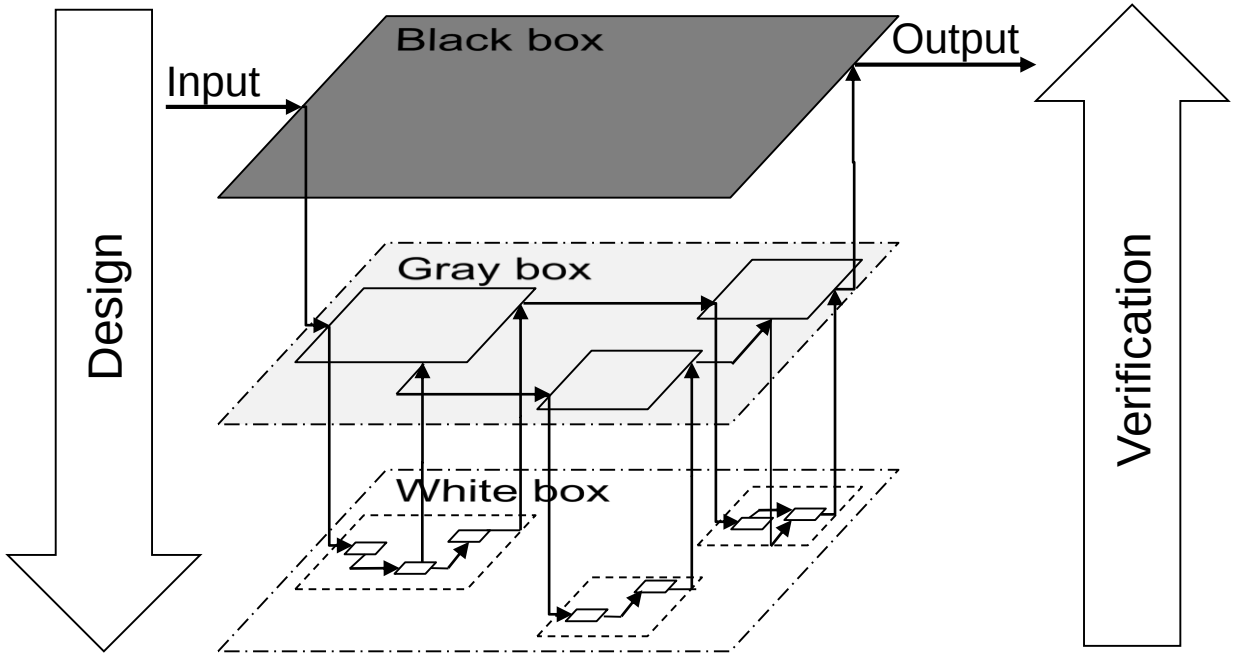
Our Approach:

1. Search for Possible Implementation using **existing and tested circuits** and **hypothetical circuits**
2. Find optimal solution(s) for satisfying requirements
3. Automatically build and execute simulation model
4. Build rapid prototype
5. Automatically analyze performance of simulation model and prototype
6. Iterate till design converges

Problem: How to build hardware prototypes as easily as simulation models?

DLR Core Avionic Testbed (CAT): Motivation

- Start from highest level functional requirements that are consistent with stakeholder expectations and problem definition
- Functions usually have to be verified by test, to demonstrate that the stakeholder needs are met
- High level functions are verified by end2end tests
- → Core Avionics Testbed (CAT):
  - Environment for functional and electrical verification (system and subsystem level)
  - Reduced effort by automation & MBSE
  - System tests early in the project schedule
  - Support of AIV activities and OPS training



Slide by Michael Jetzschmann (DLR Institute of Space Systems) with modifications

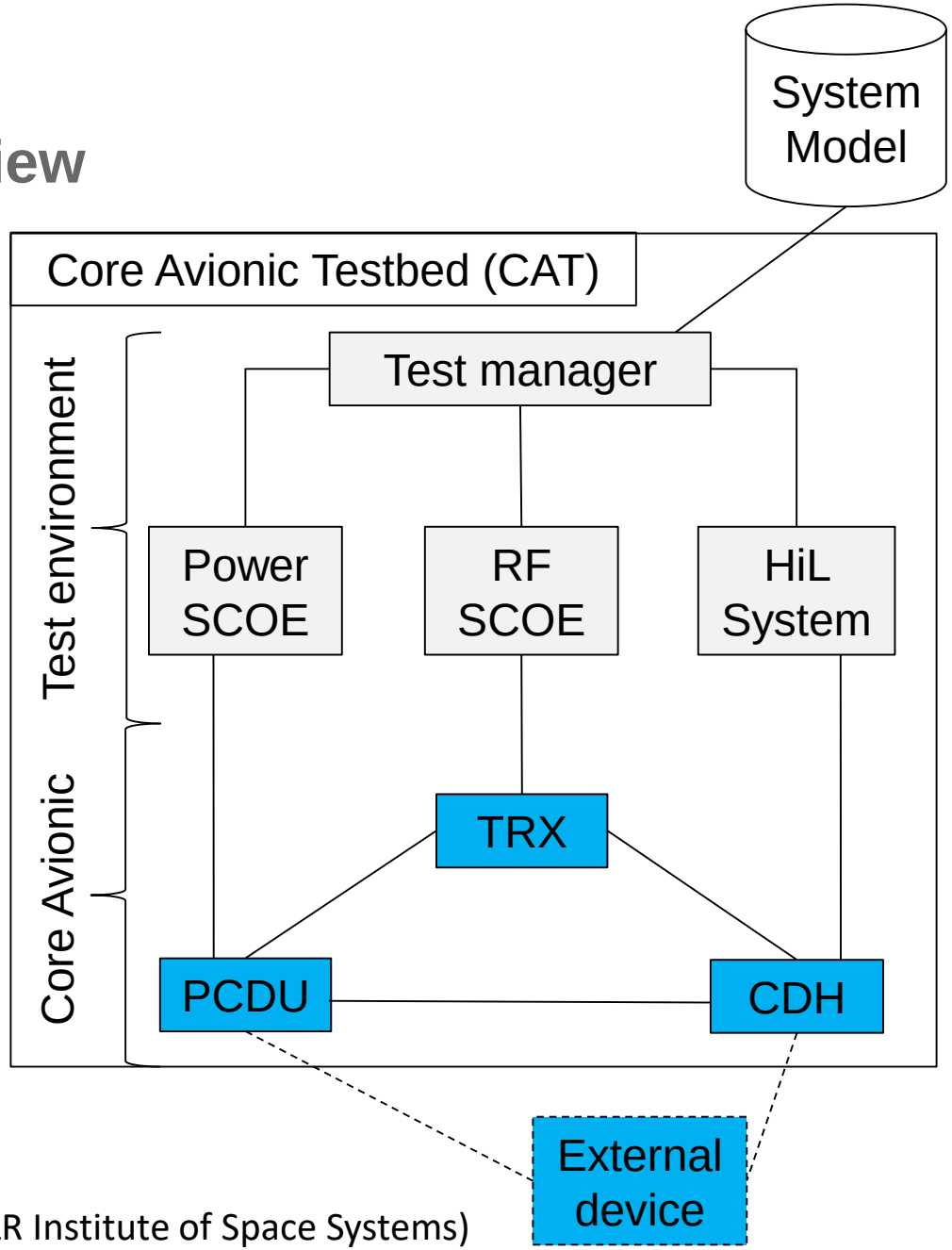
DLR Core Avionic Testbed (CAT): Goals

- Modelling of mission scenarios and environment
  - Orbit:
    - Sun & eclipse
    - Disturbances
    - Magnetic field
    - Ground station passes
    - Target area passes
  - Attitude
    - Simulate sensor output
    - Simulate “virtual” S/C attitude considering disturbances & actuator control commands
- Failures can be induced (e.g. corrupted sensor data)
- Provide infrastructure to collect, store and analyze measuring data
- Automation
  - Test cases are generated out of the system model
  - Configuration is deployed fully automated to all connected components (system and test environment)
  - Test scenarios can be replayed with automated verification of the results
  - Verification results are pushed back to the system model
  - Test reports can be generated automatically
- Models and physical devices should be exchangeable (AIV support)
- Provision of remote access

Slide by Michael Jetzschmann (DLR Institute of Space Systems) with modifications

DLR Core Avionic Testbed (CAT): Overview

- Test environment:
  - Power SCOE
    - Solar array simulator
    - Battery simulator
  - RF SCOE
    - Ground station simulator
  - HiL System
    - Sensor & actuator simulator
  - Test manager
    - Managing instance
- Core avionic:
  - Power Conditioning and Distribution (PCDU)
  - Transceiver (TRX)
  - Command and Data Handling (CDH)
- External devices
  - P/L instruments



Slide by Michael Jetzschmann (DLR Institute of Space Systems)

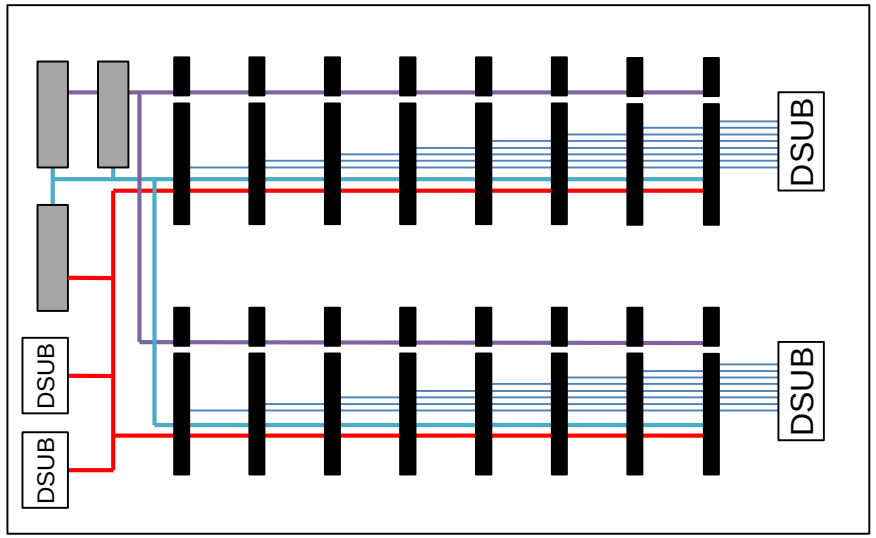
Rapid PCDU Prototyping in CAT: PCDU Modular Breadboard

Baseboard for 16 Breakout Boards

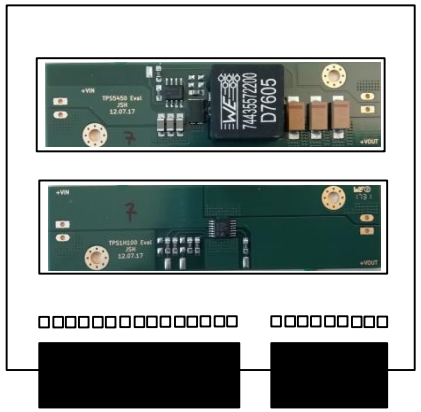
Analog and Status/Control Signals of all Modules

Secondary Power Supplies

Main Bus Input/Output



Breakout Board



DCDC & LCL Module

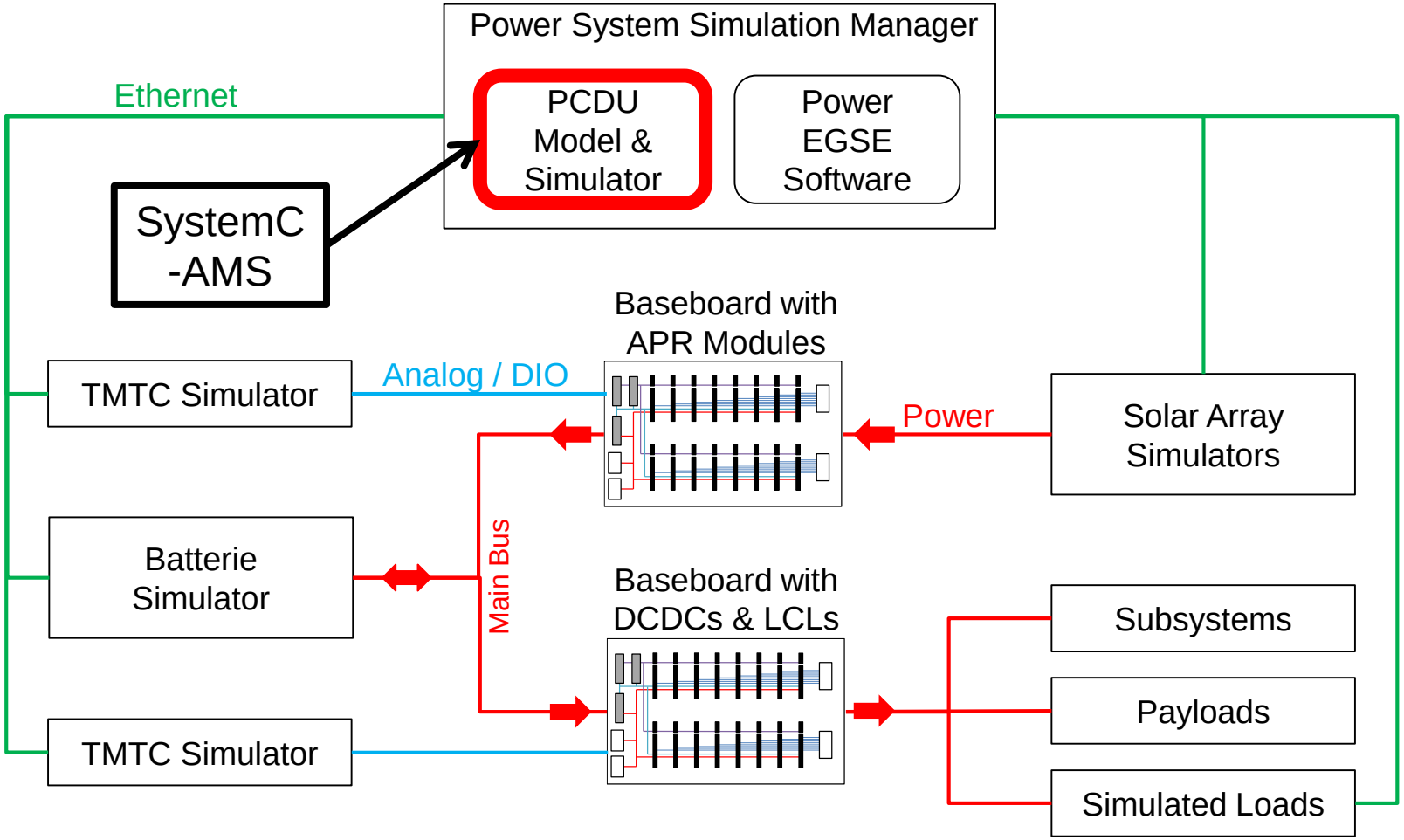
Control, Status, and Measurement Signals

Bus Voltage and Module Output

- Connecting a number of evaluation boards to fully functional PCDU modules
- Control and telemetry circuitry decoupled from power electronics (LCLs/DCDCs)
- Rapid assembly of a functional unit and easy reconfiguration
- Old revisions of modules can be swapped easily with new designs
- Baseboard can be configured as APR or distribution module

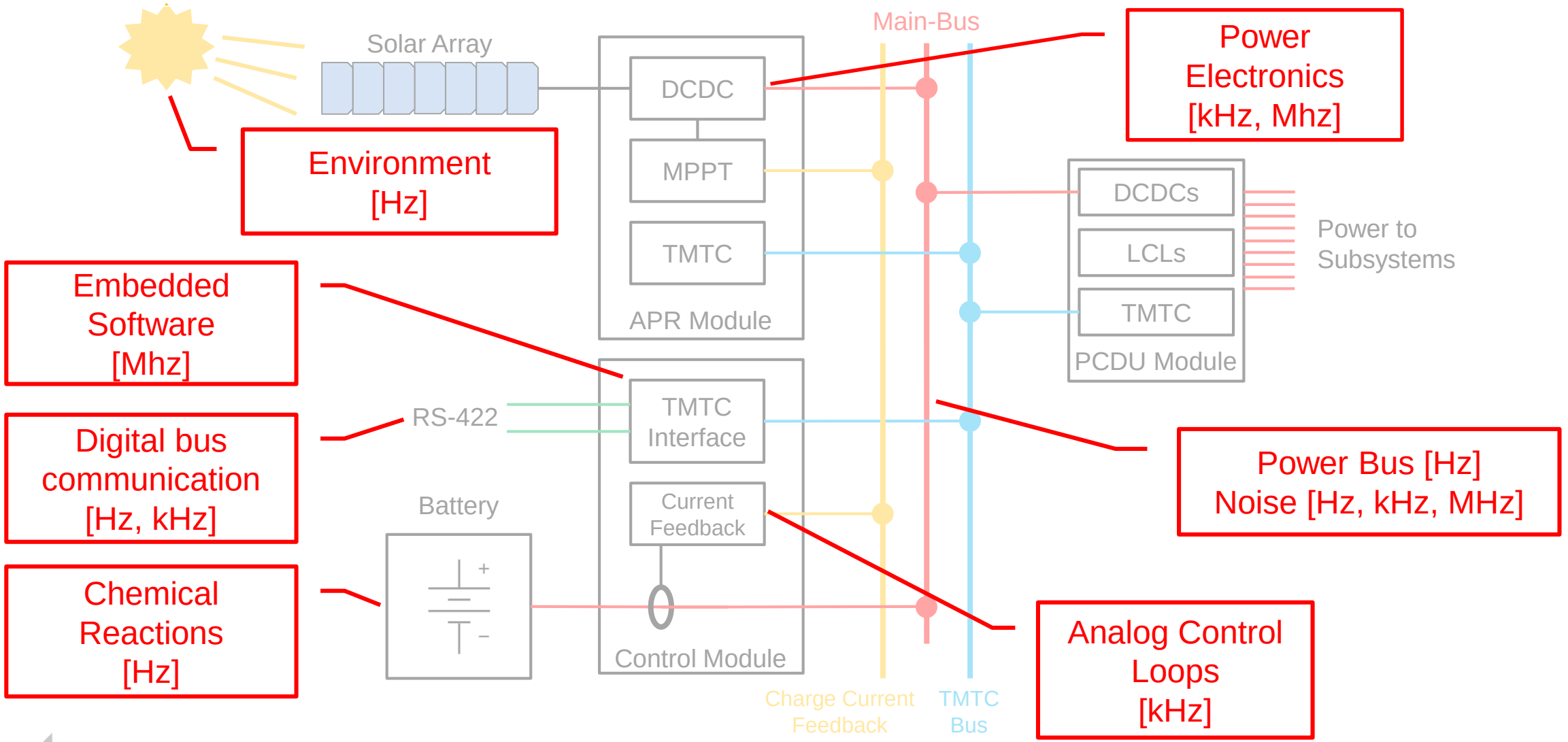
LCL: Latching Current Limiter
APR: Array Power Regulator

PCDU Modular Breadboard – HIL Setup



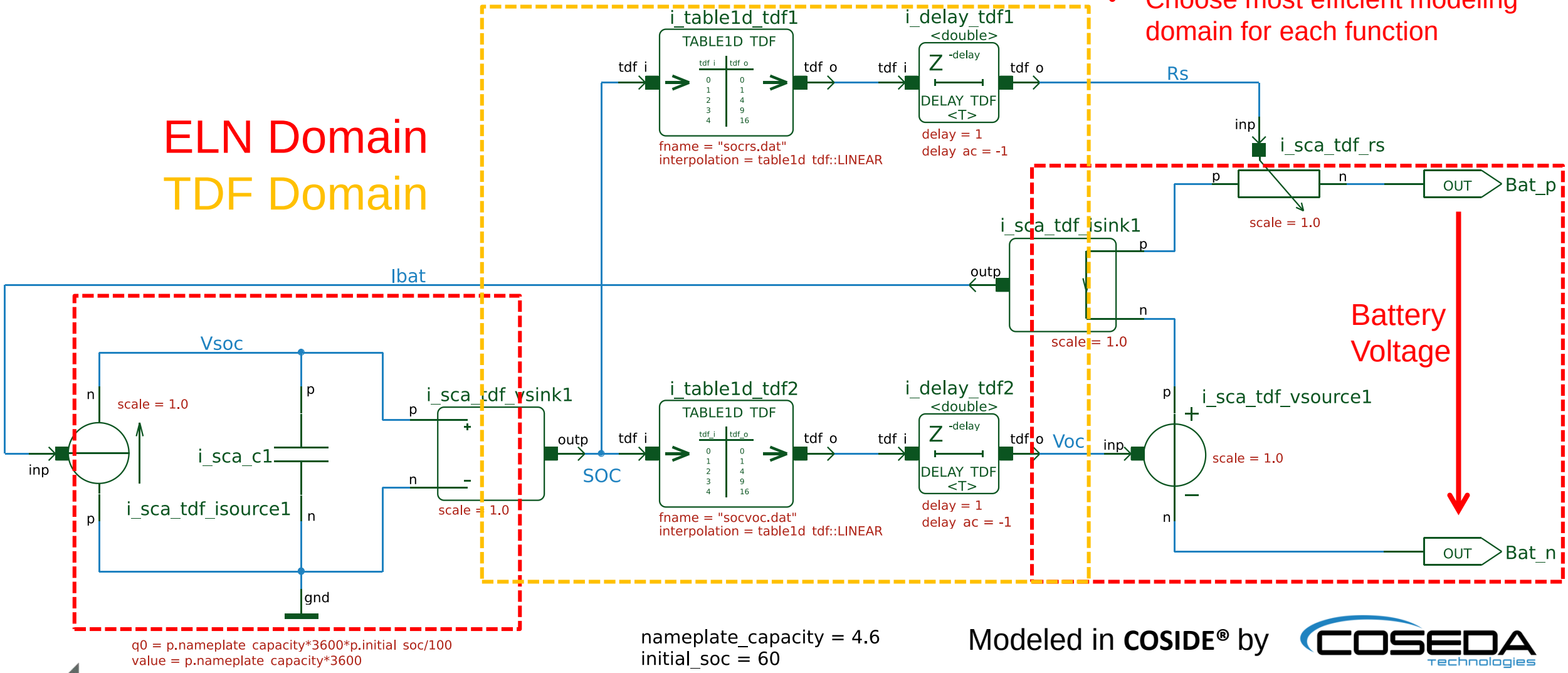
Realtime simulation of PCDU control logic and not existing components:
Modular breadboard telemetry is equal to system in full configuration!

Scalable Power System – Models and Timescales



SystemC-AMS Model of a Battery Cell

- Auto-generation of repetitive electrical circuits with constructors
- Choose most efficient modeling domain for each function



Modeled in **COSIDE®** by COSEDA technologies

Strong Points for SystemC-AMS in S/C Modelling

- Suitable for model generation from scripting language
- Cyber physical system modelling (hardware, software and environment)
- Model execution without the need of external tools and licenses
- Blackbox models (IP cannot be extracted from model binary)
- Ability to handle a wide variety of timescales in a single model
- Build abstract and fast models for analog circuits
- Models can be executed on embedded systems (ARM processors)
- High quality modeling IDE COSIDE®

Summary

- First steps of demonstrating the use of SystemC-AMS for S/C system modelling
- Introduction of an automated design process for bespoke design of PCDU for science missions
- Application of the platform based design philosophy for simulation models, hardware and software components

Future Work:

- Integrate SystemC-AMS with requirements management and rapid hardware prototyping solution
- Automation of the PCDU design cycle from requirements to PCDU schematics
- Build a comprehensive database of power circuits with associated sim. models schematics, and layouts
- Demonstrate this approach for differently sized power systems and mission scenarios