Distributed MILS Project
Work-in-Progress

Rance DeLong
Distributed MILS (D-MILS) Project

- European Commission FP7
  - Project #318772
  - December 2012 – December 2015

- D-MILS Project Consortium Partners
  - Fondazione Bruno Kessler – Italy
  - Fortiss – Germany
  - Frequentis – Austria
  - LynuxWorks – France
  - RWTH Aachen University – Germany
  - The Open Group – United Kingdom – Lead
  - TTTech – Austria
  - Université Joseph Fourier – France
  - University of York – United Kingdom
Short MILS Review

- MILS is a **component-based approach** for the construction, assurance, and certification of **dependable systems** that encourages a commercial marketplace of off-the-shelf high-assurance components.

- MILS can be understood as a **two phase approach**:
  - **Architecture**
    - Abstract **policy architecture** represented with “boxes” (operational components) and “arrows” (interactions).
    - System **purpose** is achieved by behavior of the operational components and their interactions.
    - Assumption: the architecture **will be strictly enforced**.
  - **Implementation**
    - A robust resource-sharing platform composed of **MILS foundational components** creates strongly isolated “exported resources”.
    - Components individually developed and assured according to standard specifications.
    - Components compose “additively” to form a distributed trusted sharing substrate, the **MILS Platform**.
The architecture expresses an interaction policy among a collection of components.

Circles represent architectural components (subjects / objects).

Arrows represent interactions.

The absence of an arrow is as significant as the presence of one.

Components are assumed to perform the functions specified by the architect (trusted components enforce a local policy).

Suitability of the architecture for some purpose presumes that the architect’s assumptions are met in the implementation of the architecture diagram.

Trusted Subject

This component has no interaction with any other.
Assumptions Implicit in the Architecture
Represent Two Primitive Policies

1. Isolation

C1 → C2

These components / connections have no interaction with each other

2. Information Flow Control

C3

C2

C1 → C2

Only explicitly permitted causality, data flow, or interference, is permitted. The architecture permits this flow. Only C1 or C2 can cause the flow, C3 can not. The flow is directional and intransitive.
The MILS Platform: Resource-Sharing Components

Exported Resources

Additive Composition

additive compositionality – e.g.,
Partitioning Kernel ⊕ Partitioning Net
= Partitioning (Kernel + Net)

MP = MILS Platform
+ D-MILS includes limited versions of Net(work) and Con(sole)
* D-MILS does not include MILS FS, EA and Aud components
MILS Platform – Provides Straightforward Implementation of Policy Architecture

Architecture
Validity of the architecture assumes that the only interactions of the circles (operational components) is through the arrows depicted in the diagram.

Implementation
SK, with other MILS foundational components, form the MILS Platform allowing operational components to share physical resources while enforcing Isolation and Information Flow Control.
Inspiration for Distributed MILS: Policy architecture deployment spanning nodes

- Node Hardware
  - SK
  - MNS
- Subjects
- Foundational Plane
  - SK ⊕ MNS
  - Node Hardware
  - Node Hardware
Distributed MILS

- A single policy architecture may span multiple D-MILS nodes expressed in declarative MILS-AADL
- Guarantees similar to a single MILS node: isolation, information flow control, determinism
- Determinism over network could be achieved in various ways – in D-MILS we use Time-Triggered Ethernet (TTE)
- Must configure and schedule the network and the processors of the nodes coherently
- Support verification of architectural properties, presentation of assurance case, and generation of configuration with integrated automation with the greatest practical use of existing verification technology
D-MiLS Technology and Application Areas

- Graphical and Declarative Languages (front-end)
  - Architecture Analysis and Design Language, MILS extended subset (MILLS-AADL)
  - Goal Structuring Notation (GSN)

- Integration of GSN and MILLS-AADL
  - Structure of GSN assurance case informed by information gleaned from MILS-AADL model

- Representation Semantics and Transformations
  - Semantics preserving transformation between front-end languages and intermediate and back-end languages of the analysis tools

- Compositional Verification
  - Reduce verification of system to independent verification of its parts
  - Properties to verify and appropriate verification strategies and tools

- Compositional Assurance Cases
  - Modular GSN - Rely / guarantee argumentation

- Configuration Compiler
  - Generate configuration information for D-MiLS nodes and connecting TT network infrastructure
  - Configuration constrained by actual physical resources available and other semantic analyses

- D-MiLS Platform
  - TTEthernet – Distributed MILS network drivers and configuration
  - LynxSecure separation kernel with MILS Networking Subsystem (MNS)

- D-MiLS Industrial Demonstrators
  - Smart Micro Grid – fortiss
  - Voice Services – Frequentis

© 2013 D-MiLS Project
D-MILS Technology Areas

- Architecture Analysis and Design Language
- Intermediate Languages
- Verification Framework
- Assurance Framework
- Compositional Assurance Case
- Graphical & Declarative Languages
- Compositional Verification

Integration GSN & AADL
Configuration Compiler
D-MILS Platform

Pre-existing products
LSK TTE

Extended Separation Kernel
Ext. Time Triggered Ethernet
Distributed MILS Technology Elements

- **MILS-AADL ext’d subset**
- **Representations & Transformations**
- **Configuration Compiler**
- **SK Config’n**
- **Resource Inventory**
- **TTE Config’n**
- **Separation Kernel**
- **MCS**
- **MNS**
- **TTE Ethernet**
- **fortiss Smart Microgrid**
- **Frequentis Voice Services**

**Tools**
- GSN Assur. Case
- Verification Evidence
- Verification System
- Configuration Compiler
- MILS-AADL ext’d subset
- Representations & Transformations
- SK Config’n
- Resource Inventory
- TTE Config’n
- Separation Kernel
- MCS
- MNS
- TTE Ethernet
- fortiss Smart Microgrid
- Frequentis Voice Services

**Colors and Symbols**
- Tool Input / Output Artifact: Cyan
- Tool: Yellow
- Modified or new component: Pink
- Application demonstrator: Gray
- Artifact generation/use: Orange
- Resource availability: Red
- Technology Application: Green

**Automation**

- System Purpose
- System Properties

**D-MiLS Platform**

© 2013 D-MiLS Project
Distributed MILS Platform – MILS nodes with deterministic communication

A Distributed MILS Platform:

Enables:

Realization of deterministic distributed MILS architectures

SK ⊕ MNS
Foundational Plane

Node Hardware  Node Hardware  Node Hardware  Node Hardware  Node Hardware
D-MILS System Assurance Case Structure

- Compose assurance cases modularly using Assume-Guarantee Reasoning
- D-MILS System assurance requires the validity of three sub-cases
- Assumptions from D-MILS System assurance case become obligations on the sub-cases

---

D-MILS System

D-MILS System High-Level Assurance Argument

Policy Architecture

MILS Platform

Environment

Sub-case

Sub-case

Sub-case

Assume

Guarantee

Assume

Guarantee

MILS Platform Assurance Argument

Policy Architecture Assurance Argument

D-PM Goals

Env Goals

PA Goals

D-MILS Project WiP © 2013 D-MILS Project
MILS Platform Assurance Case Structure

- The D-MILS Platform is composed of three major subsystems: MSK, MNS, MCS
- Assumptions from D-MILS Platform assurance case become goals for the components
- Assured Goals from component assurance cases become evidence for D-MILS Platform sub-cases
- Ground evidence provides the ultimate justification for the assurance case
Demonstrator: fortiss Smart Microgrid
Demonstrator: Frequentis Voice Services

cwp... controller working position
rce...radio control equipment
r-rce...remote rce
c-rce...center rce
swim...system wide information management