A Data Centric Approach for Modular Assurance

The Fourth Layered Assurance Workshop
December 6th 2010

The Real-Time Middleware Experts

Gabriela F. Ciocarlie
Heidi Schubert
Rose Wahlin

Mixed Criticality Systems

- Any system that has multiple assurance requirements
  - Safety, at different assurance levels
  - Security, at different assurance levels
- Example: Unmanned Air Vehicle
  - Flight control is safety critical
  - Payload management is mission critical
- Ideally a system is built from components each with their own assurance requirements

The Challenge

- Design a modular plug-and-play architecture to reduce cost and reuse components
- Components must interact
  - The behavior of one component can affect another
  - It can be advantageous to have components at different criticality levels exchange data
  - Once a component interacts with another, then the whole system must be certified, not the individual components

The Solution

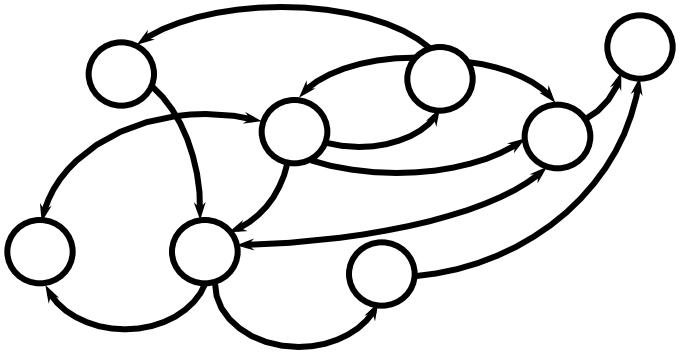
- Move from a component-interaction model to a **data-centric model**
- The data-centric model defines the data types and attributes in the system
- A component complies with the data model in terms of data it sends and receives
- This **decouples** applications

Agenda

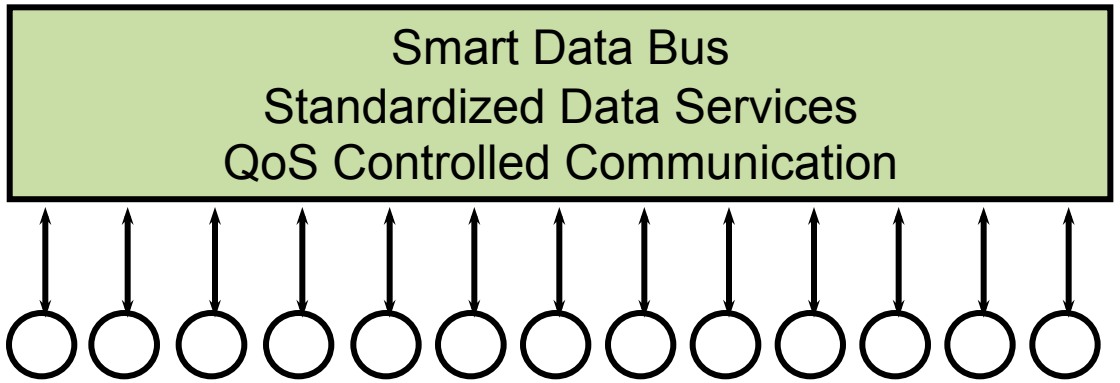
- Introduction
- Data Centric Architecture
  - Modularity
  - Separation Kernels
  - Anonymous Publish-Subscribe
  - OMG Data Distribution Service
- Feasibility Study
- Conclusions
- Future Work

The Modular Approach

Monolithic Approach
- Certify whole system
- Connection oriented
- Tightly coupled
- Hard to evolve



Modular approach
- Certify components
- Data oriented
- Loosely coupled
- Evolvable



The Data Contract

- First, all exchanged data in the system is defined
- Next, data characteristics are defined
  - For example “airspeed” is flagged as flight critical
- Then components define data delivery attributes
  - A flight critical component specifies data rate that flight critical data must be delivered
- This creates a **“data contract”**

Data Centric Approach for Mixed-Criticality Systems

- Data contract includes
  - Data type
  - Name
  - Quality of Service
- Validation
  - Component validation – does it conform to the data model
  - System validation – is there a producer at correct assurance level for each required data

Realization of a Mixed-Criticality System

- Separation kernels
  - Guarantees isolation of components
  - Controls data flow
- Anonymous publish-subscribe
  - Used to implement the data model and distribute data

Separation Kernels

- Partial solution for mixed-criticality systems certification
- Isolation and Control
  - Each guest operating system (OS) **runs in its own partition**
  - Each guest OS is **isolated** over both **time and space**
  - Information flows are **tightly controlled**
  - Components can be pre-certified and composed quickly into new configurations
- **Challenge**
  - Do not address interdependency between components or interactions between components on separate computers

Anonymous Publish-Subscribe

- Effective communication architecture
- Applications simply **publish what they know** and **subscribe to what they need**
- Networking middleware provides the functionality for:
  - discovering publishers and subscribers
  - handling network traffic and errors
  - delivering the data
- **Challenge**
  - Require effort to migrate from a point-to-point component interaction model

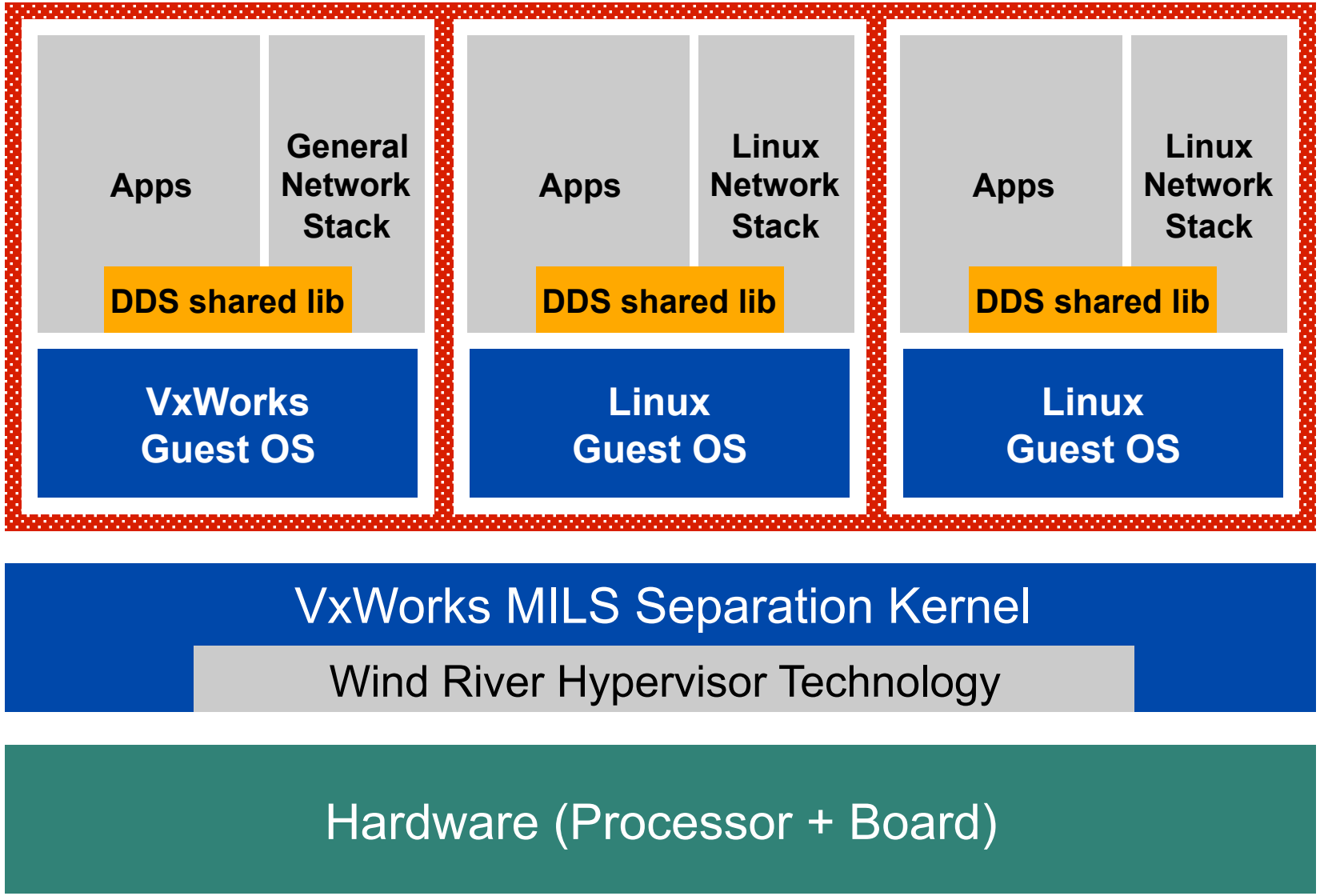
OMG Data Distribution Service (DDS)

- Data-centric publish-subscribe middleware for real-time communication
  - Strong data typing
  - Quality-of-Service (QoS) parameters
    - *e.g.,* deadlines for message delivery, bandwidth control, reliability model control, failover and backup specification, data filtering etc.
- DDS QoS parameters characterize:
  - the **data contracts** between participants
  - the **properties of** the overall **data model**
  - **real-time communication and delivery requirements** on a per-data-stream basis

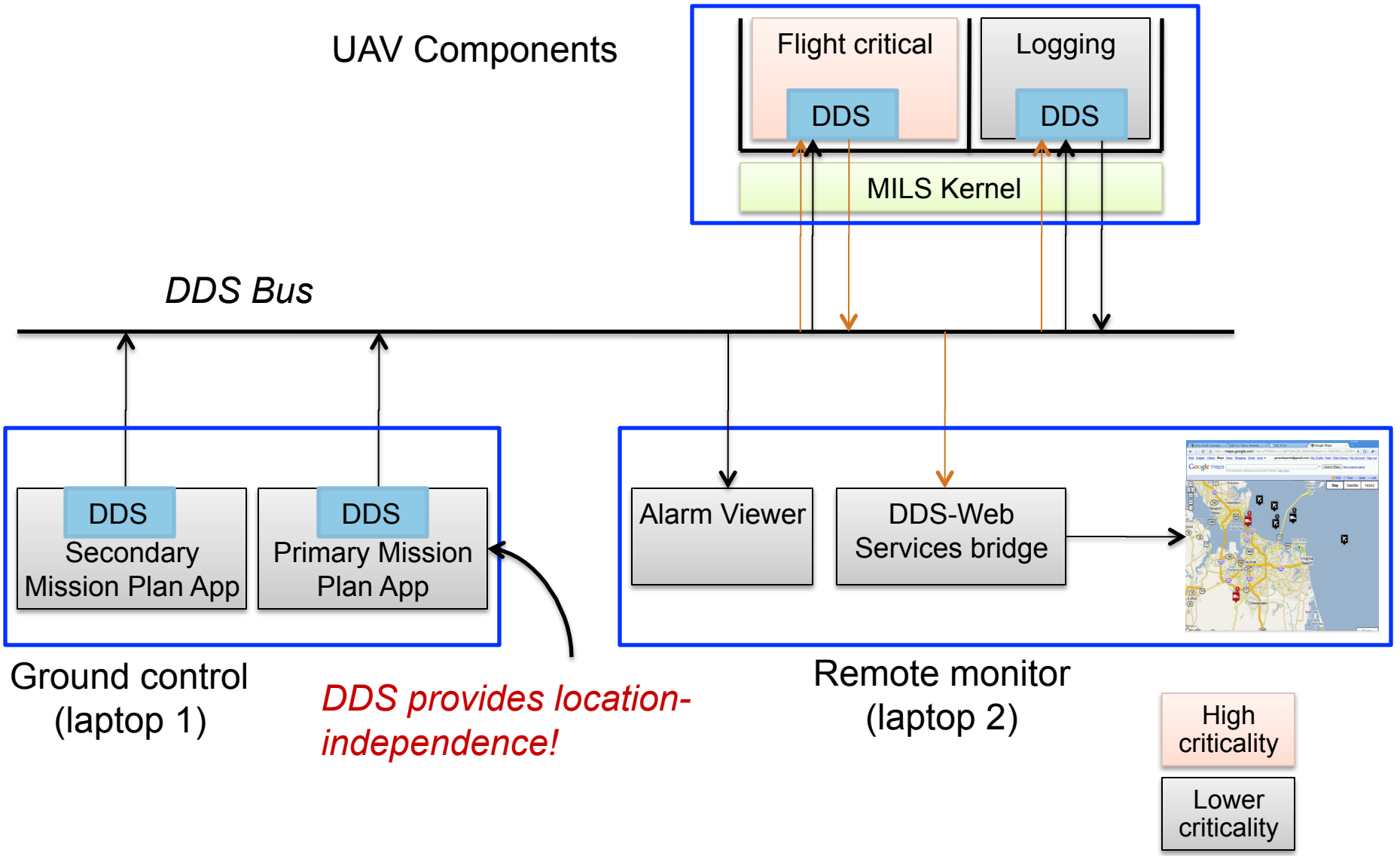
Agenda

- Introduction
- Data Centric Architecture
- Feasibility Study
- Conclusions
- Future Work

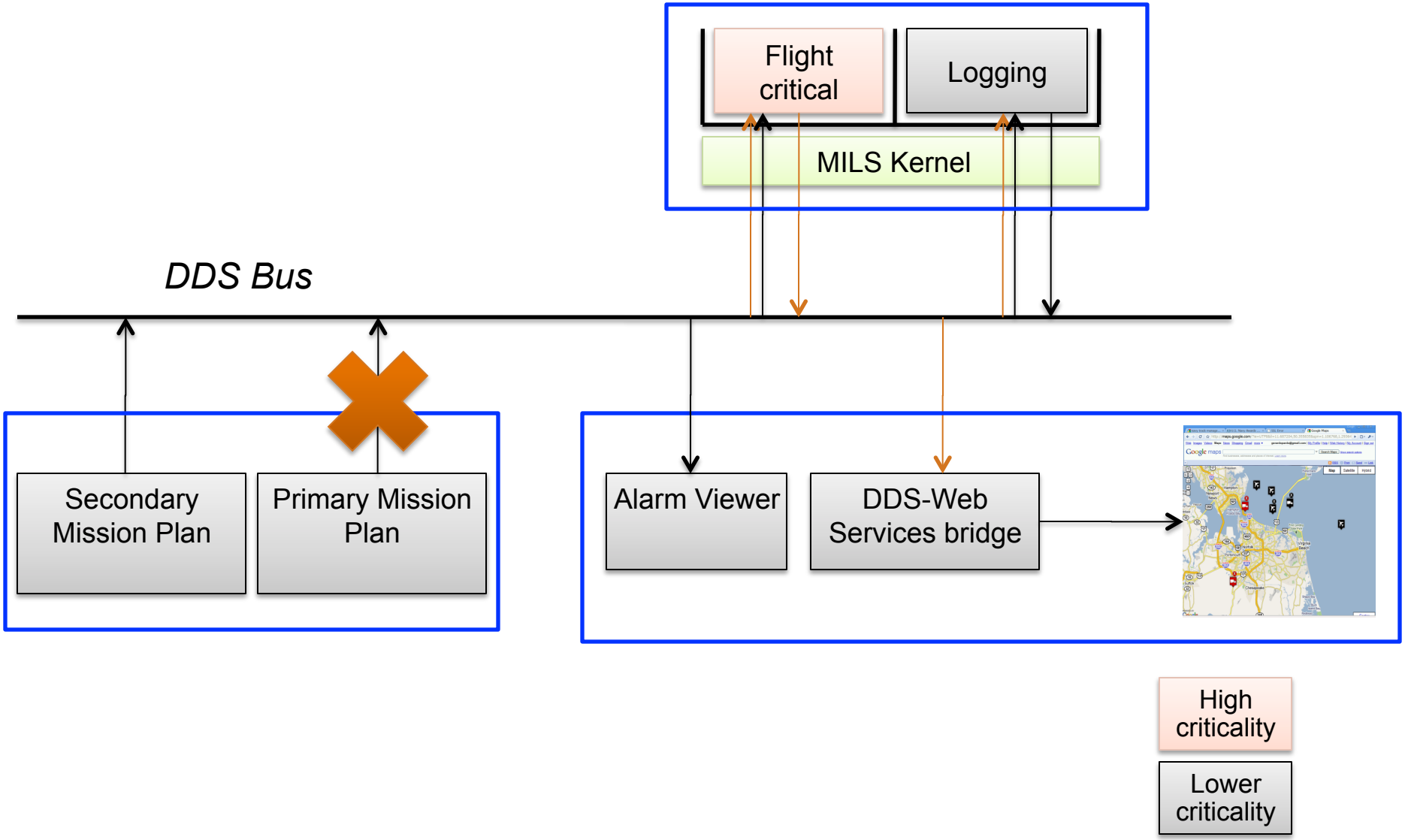
Wind River VxWorks MILS and RTI Data Distribution Service



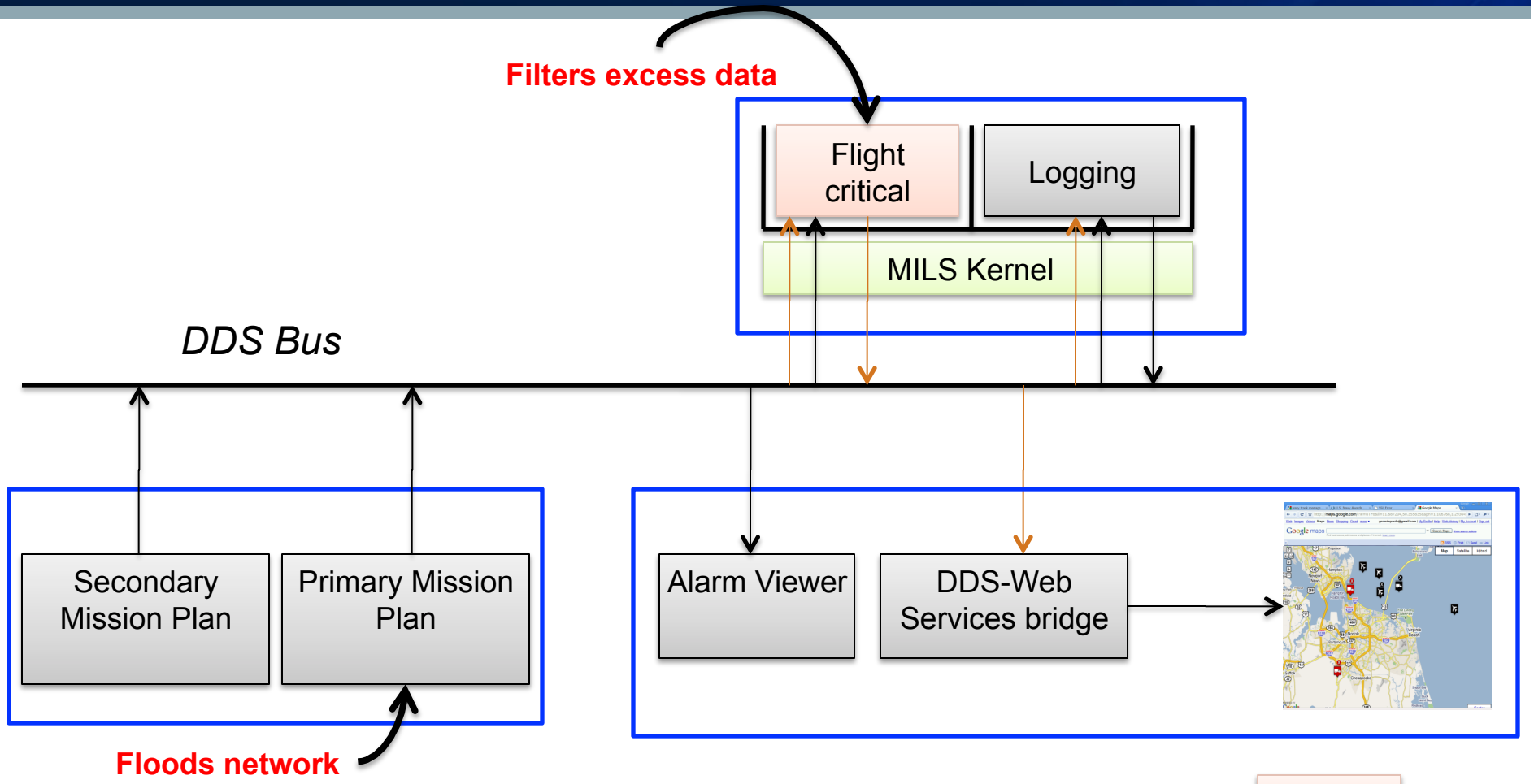
Study Overview



Scenario 1: Failover of Lower Criticality



Scenario 2: Lower Criticality Floods the Network



Conclusions

- Mixed-criticality systems certification still has a long way to go
- We can leverage:
  - Isolation and control capabilities through separation kernels
  - Modularity through a data-centric architecture
- It is possible to build mixed criticality systems that provide:
  - Modularity
  - Evolvability
  - Fault tolerance

Future Work

- Identify and analyze the characteristics of the DDS data models that lead to an efficient certification process

- Formally demonstrate the applicability of our approach to mixed-criticality systems

Acknowledgments

- **Wind River** – provided their MILS platform as well as for valuable feedback
- **United States Air Force** - contract FA8650-10-M-3025
  - The content of this work is the responsibility of the authors and should not be taken to represent the views or practices of the U.S. Government or its agencies.

Thank You