

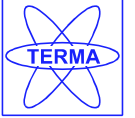
# ***Structure & Goal of Presentation***

***Gert Caspersen***  
***gec@terma.com***



## *Structure of Presentation*

1. Structure & Goal of Presentation
2. Project Objectives & Background
3. Demonstration System Highlights
4. Questions
5. System Metrics
6. Third Party Evaluation
7. Lessons Learned
8. Future Directions



# ***Project Objectives & Background***

***Gert Caspersen***

***gec@terma.com***

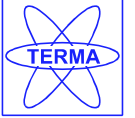
## *Onboard Operations Support Software*

### Objectives

- ❑ Potential reuse in domain of data handling systems
- ❑ Reusable data handling software components
- ❑ Demonstrate feasibility

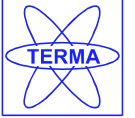
### Results

- ✓ Packet Utilisation Standard applicable to historic missions
- ✓ Reusable Ada 83 software components
- ✓ CLUSTER Data Handling System on 1750A hardware



## *Project Objectives*

- Potential **reuse** in domain of data handling systems?
- Re-engineer on **ERC32** hard real-time tools
- Applicability of concept for **Instrument Control Units**?
- **Demonstration** data handling system *and* instrument control unit
- **Economic implications** of approach for reuse



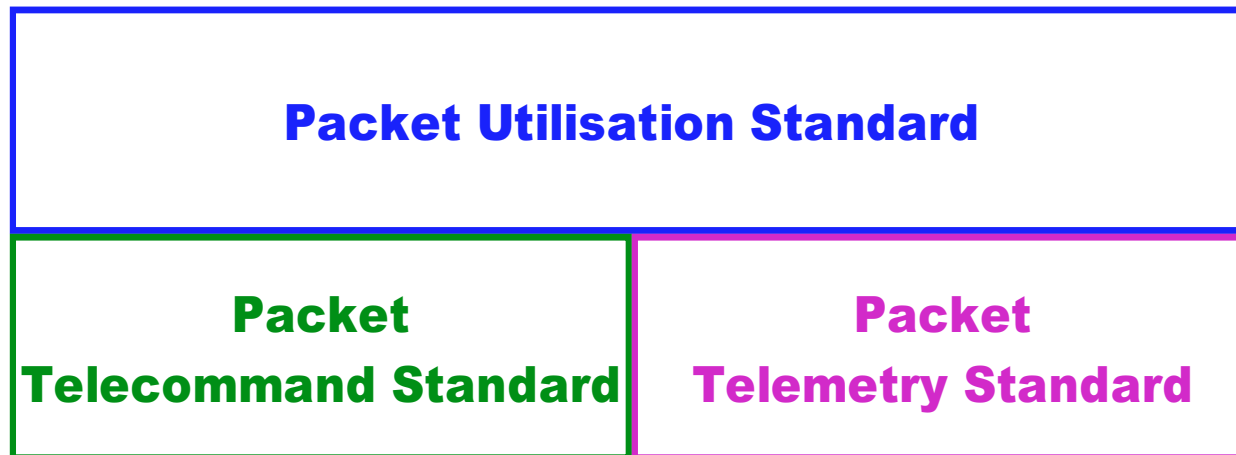
# ***Packet Utilisation Standard***

***Gert Caspersen***

***gec@terma.com***

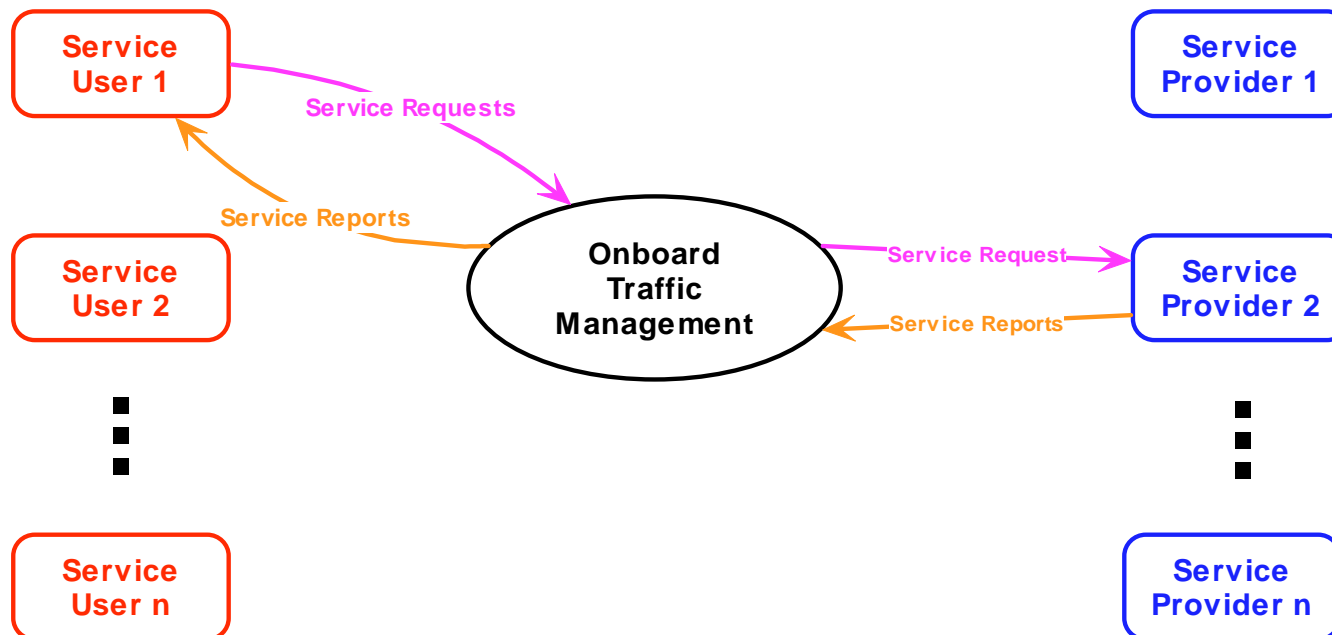
## *Packet Utilisation Standard*

- Standardises **application process layer**
- Satellite operations based on **logical model**
- Structuring of telecommands and telemetry
- Operational complexity go to space segment



## Conceptual Model

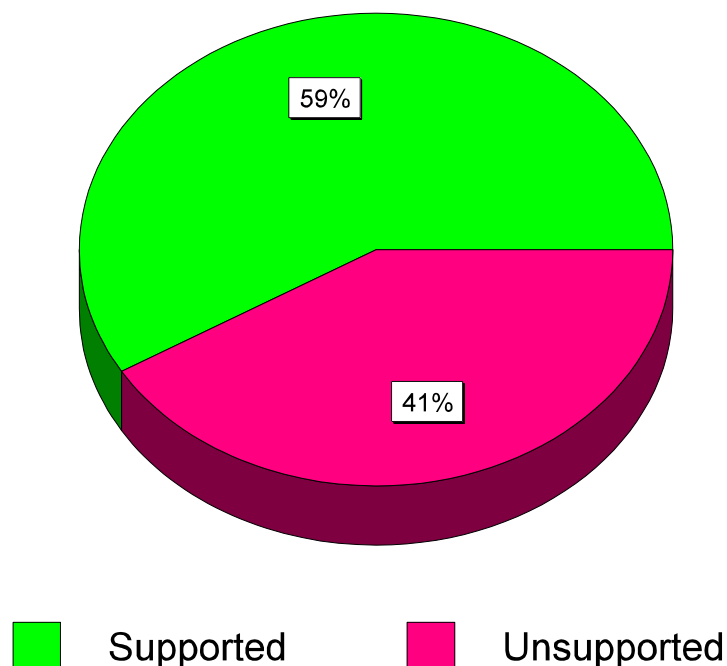
- **Service providers** (application processes) and **service users** (control centres).
- Service provider offers PUS services.
- **Service request** results in activities and **service reports**.



## *Implemented PUS Services*

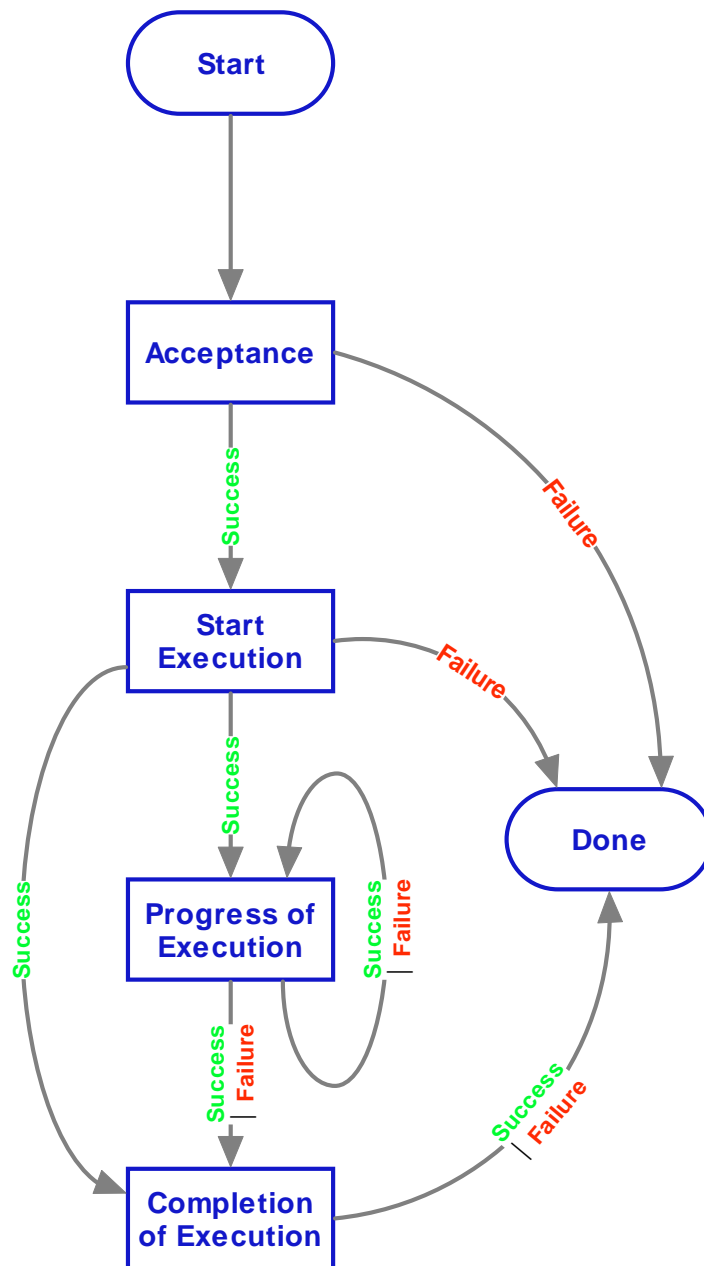
- Telecommand verification service
- Device-level command service
- Housekeeping & diagnostic data reporting service
- Function management service
- Onboard scheduling service
- Onboard monitoring service
- Onboard storage and retrieval service
- Event reporting
- Memory management
- Onboard traffic management

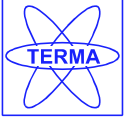
## Services



## Telecommand Verification

- Explicit verification of telecommand execution
- Verification-level dynamically controlled



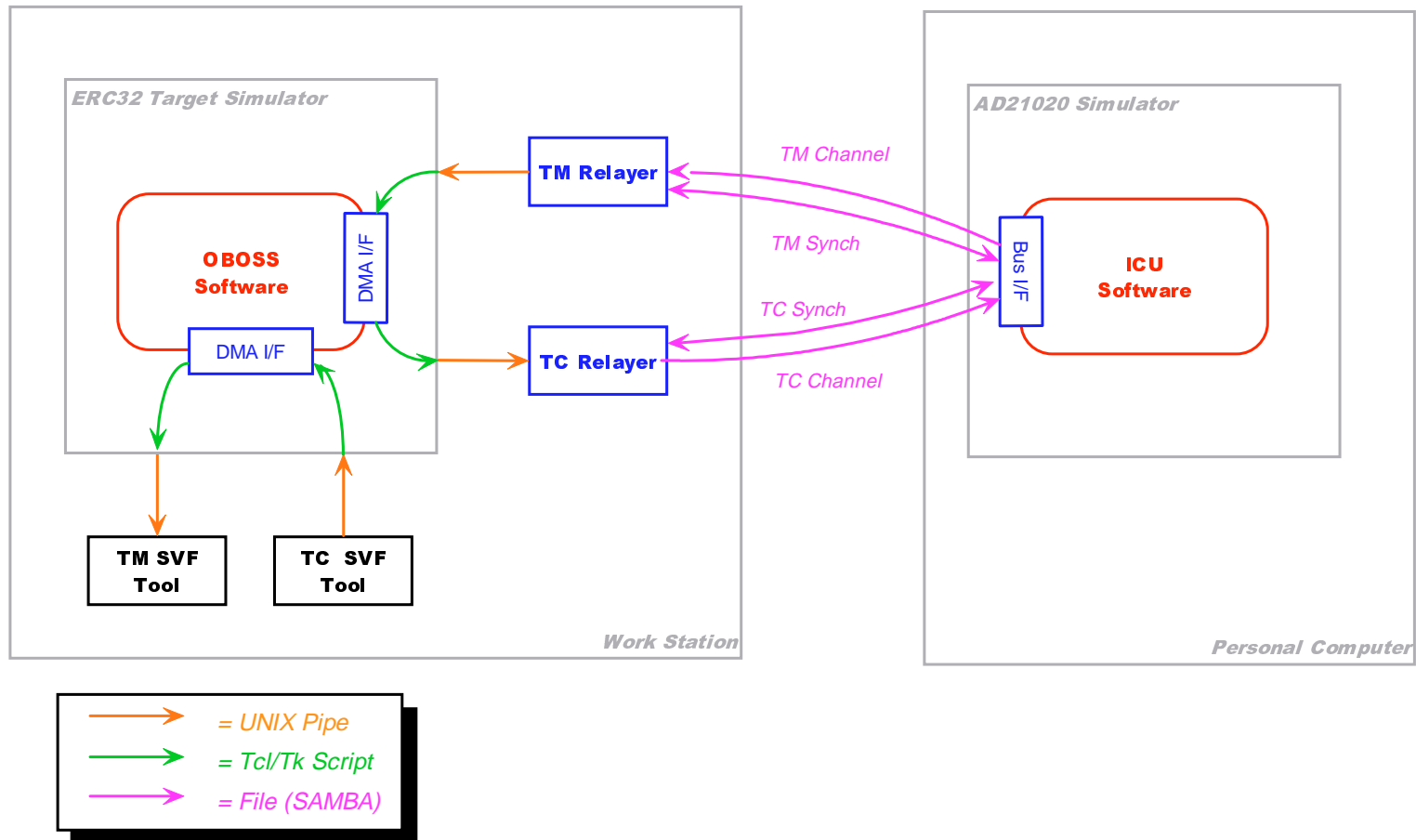


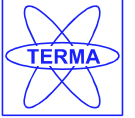
# ***Demonstration Set-Up***

***Morten Rytter Nielsen***

***mrn@terma.com***

## Demonstration Set-Up



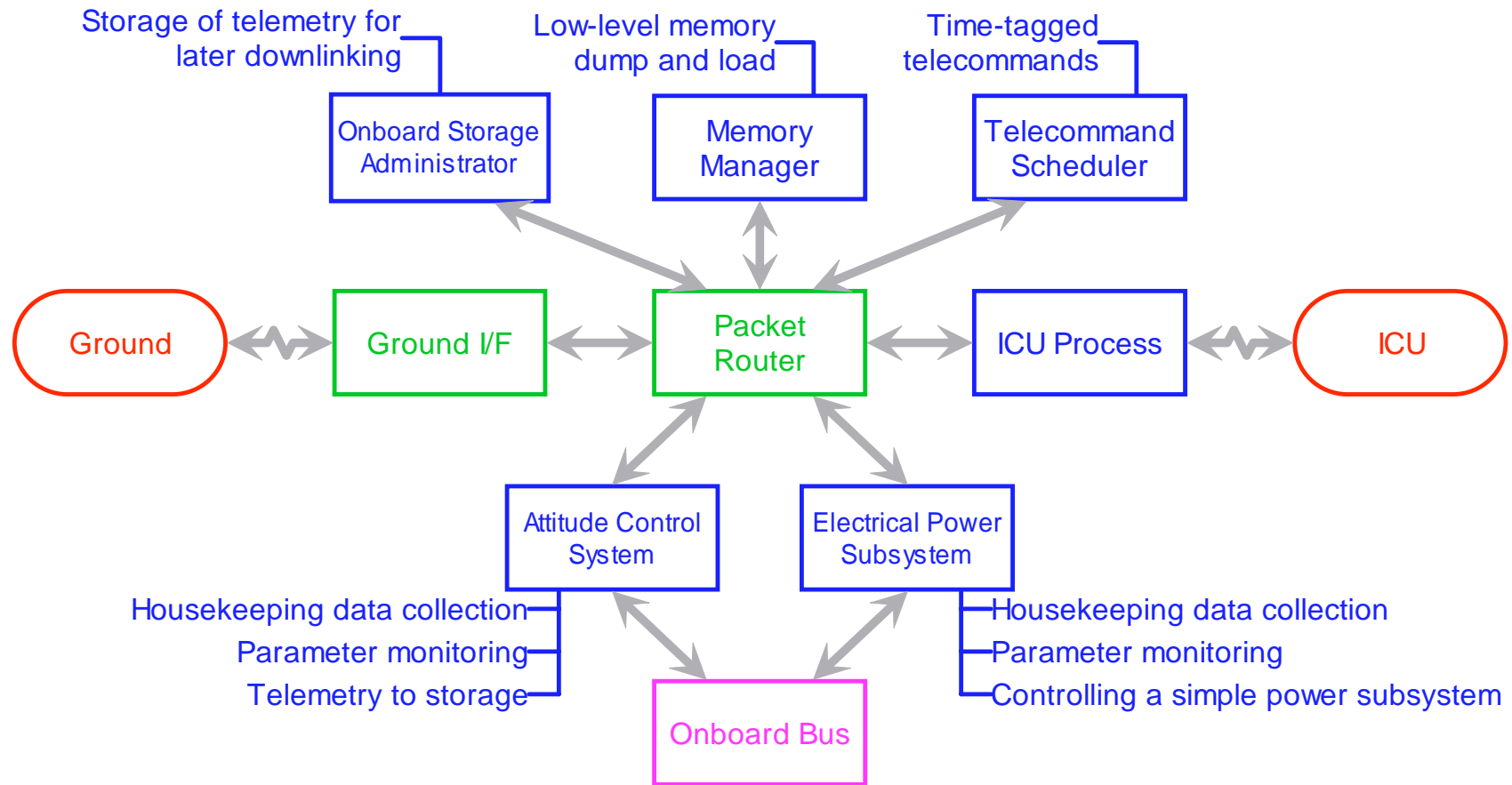


***Data Handling System  
Operational Scenario***

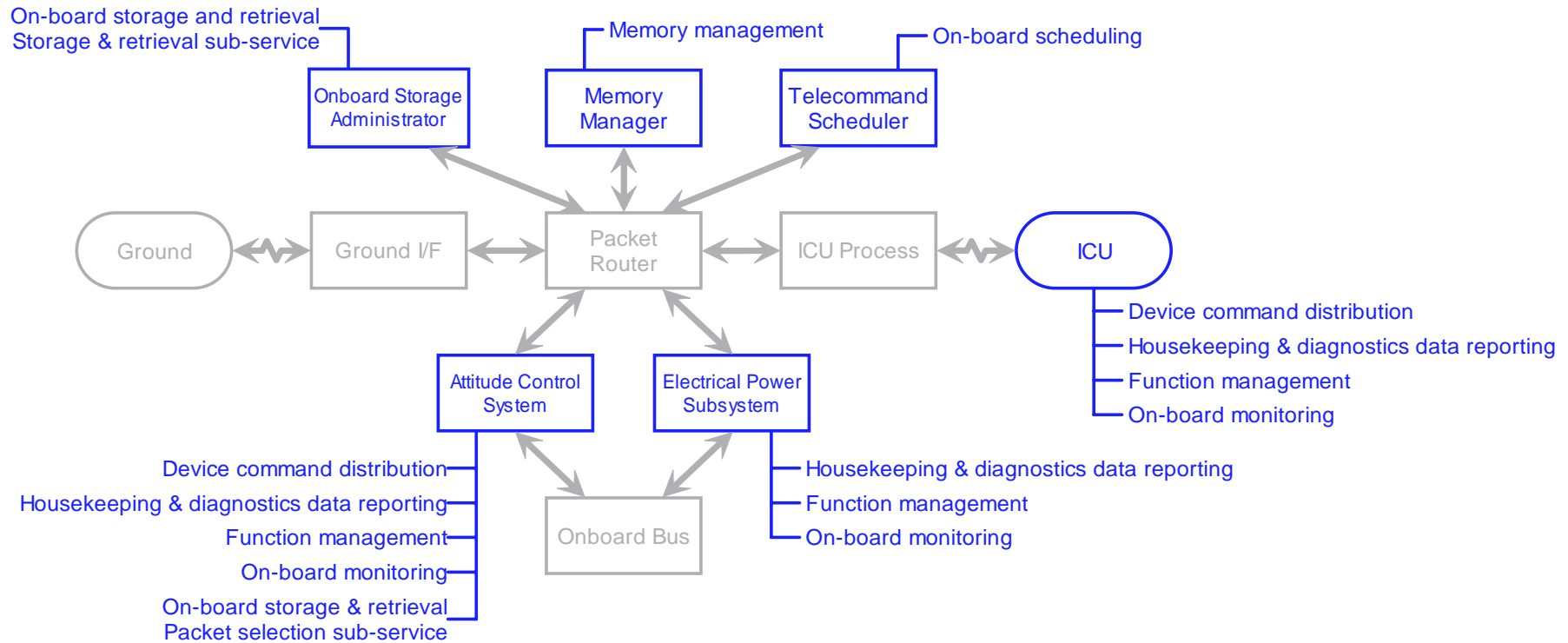
***Jan Storbank Pedersen***

***[jnp@terma.com](mailto:jnp@terma.com)***

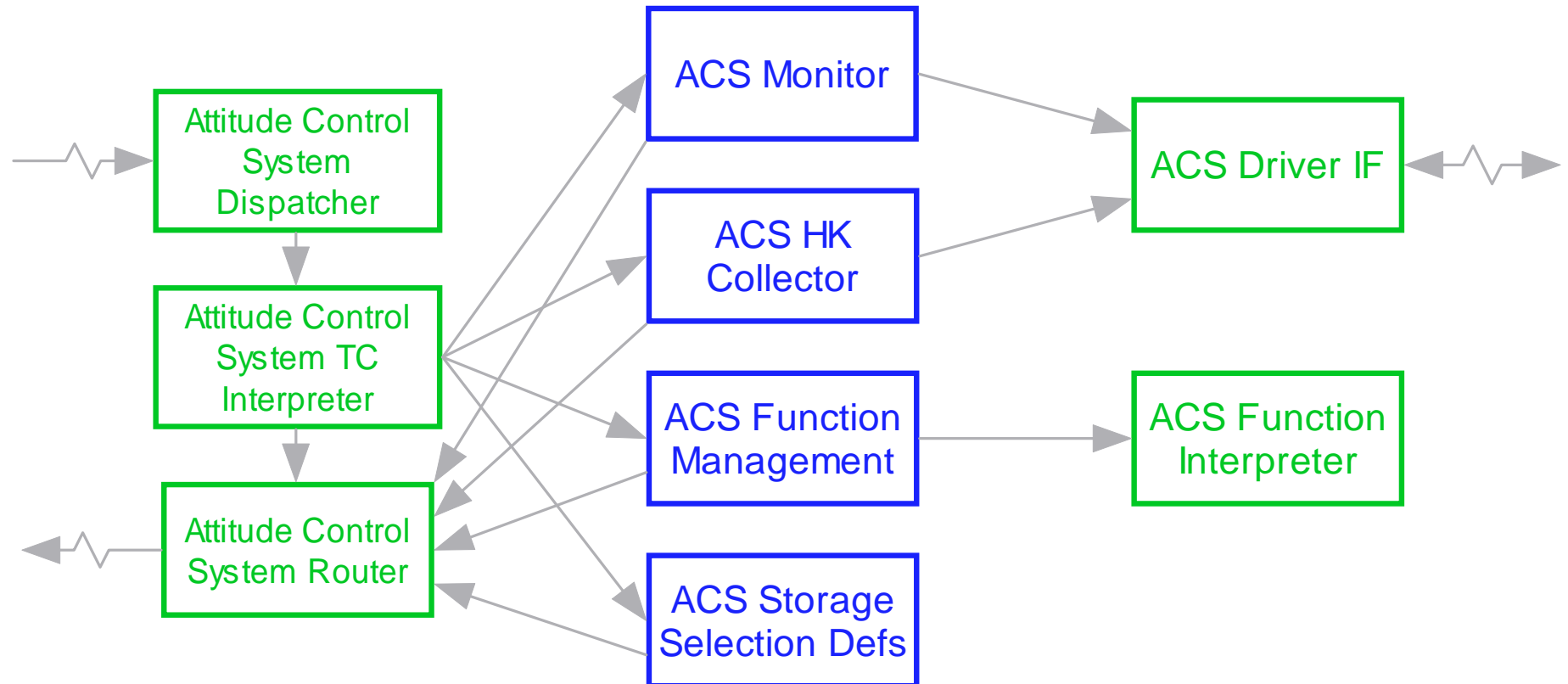
## Demonstration System



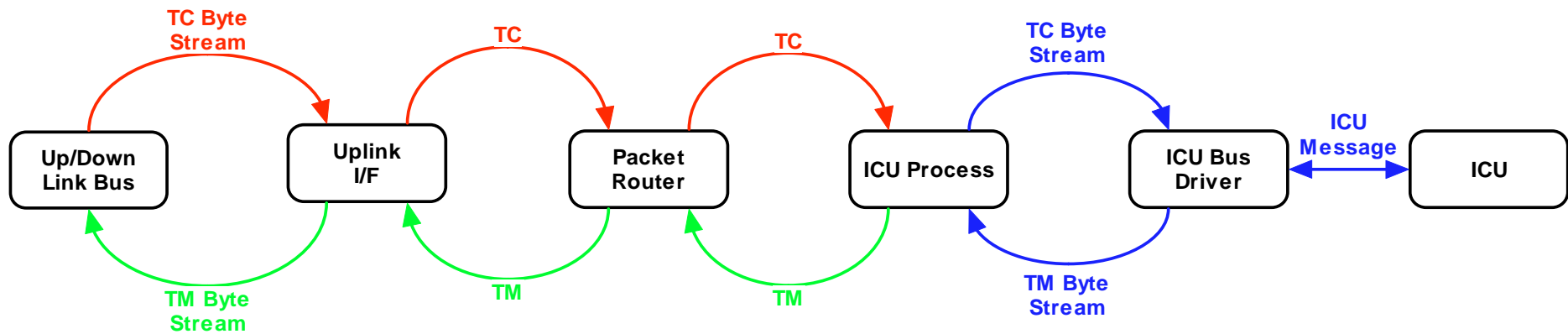
## Demonstration System Services

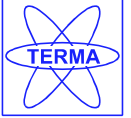


## Attitude Control System Decomposition



## Source Packet Flow





## *Telecommand Verification*

### Functionality

- Explicit verification of telecommand execution
- Command-specific control of verification level

### Demonstration

- One telecommand requesting full verification
- Several verification packets received
- Remaining data handling TCs only request verification on completion

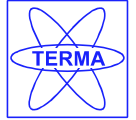
## *Housekeeping Collection and Parameter Monitoring*

### Functionality

- Periodic collection and reporting of parameter values for housekeeping
- Periodic monitoring of parameters against nominal intervals or specific values

### Demonstration

- Define and enable housekeeping collection and reporting
- Define and enable parameter monitoring
- Housekeeping reports generated
- Out-of-limit reports generated



## *Event Reporting*

### **Functionality**

- Reporting special incidents

### **Demonstration**

- Send a TC with an illegal application identifier
- Event report from Ground\_IF

## *On-board Scheduling*

### Functionality

- Time-tagged commands
- Enable and disable scheduling

### Demonstration

- Place two function management TCs on the schedule with relative time-tags
- Enable scheduling at relevant levels, and fix the release times
- TCs are being released and executed



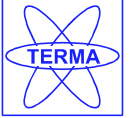
## *On-board Storage and Retrieval*

### Functionality

- Cyclic and bounded stores
- Definition of storage selection criteria for telemetry
- Enabling and disabling of storage
- Dumping telemetry from a store

### Demonstration

- Storage selection for all housekeeping packets from the attitude control system
- Enable storage
- Disable storage
- Dump and delete packets



## *Memory Management*

### Functionality

- Low-level dump and load
- Checksum

### Demonstration

- Dump part of RAM area



## *Function Management*

### Functionality

- Activate and deactivate a function
- Perform an activity (of an active function)

### Demonstration

- Activate parts of an electrical power sub-system
- Turn on units within the active parts of the electrical power sub-system

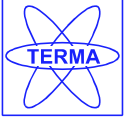
## *Autonomous Event Handling*

### Functionality

- Reaction to a parameter out-of-limit event
- Default or user-supplied behaviour

### Demonstration

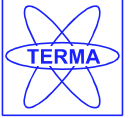
- Monitoring of total effect in electrical power sub-system
- Exceeding the allowed maximal effect
- A part of the electrical power sub-system is automatically turned off
- An event report is generated



# ***Interfacing OBOSS***

***Jan Storbank Pedersen***

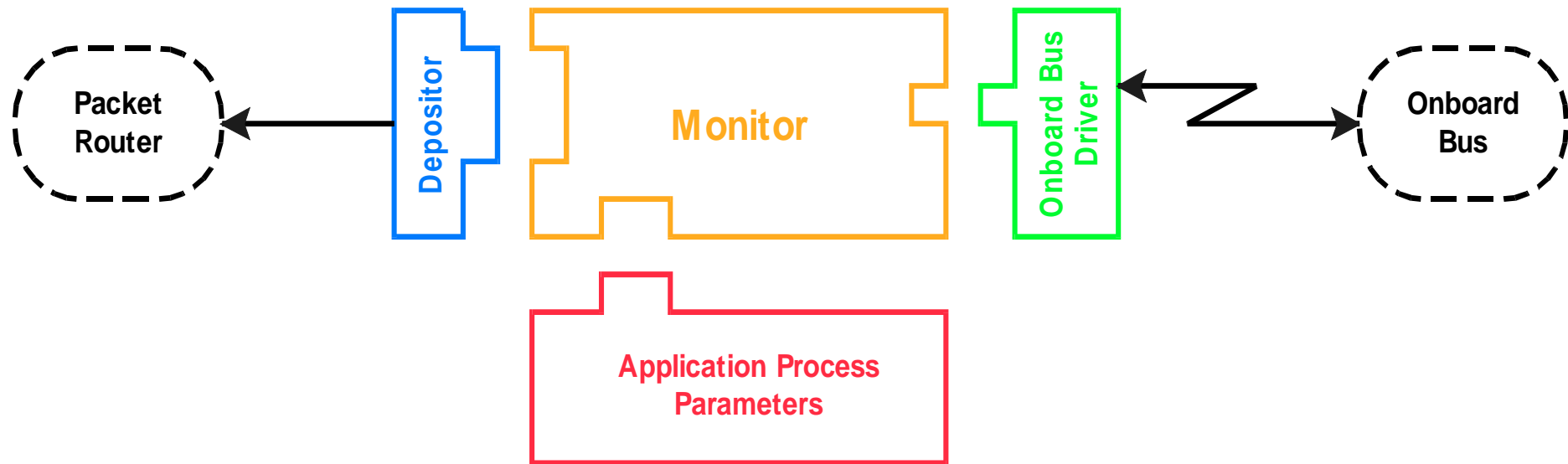
***[jnp@terma.com](mailto:jnp@terma.com)***

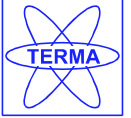


## *Issues*

- Reading parameter values for housekeeping collection and/or monitoring
- Up- and down-link of PUS packets (TCs and TMs)
- Interfacing external PUS-based systems, e.g. an ICU

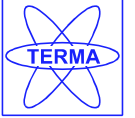
## Adapting Onboard Monitoring





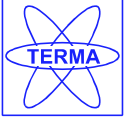
## *Up- and Down-link*

- Implemented in the on-board code by a `Serial_Bus_Simulator` package that supports "sending" (to ground) and "receiving" (from ground) arrays of bytes. Presently based on DMA.
- Currently uses a shared memory facility of the ERC32 target simulator and Tcl/Tk scripts
- Could be changed to using UART communication instead, without affecting other parts of the on-board system



## *Interfacing External PUS-based Systems*

- Very similar to up- and down-link interface, i.e. using a DMA-based bus simulator that "sends" and "receives" byte arrays
- Could easily use a UART instead



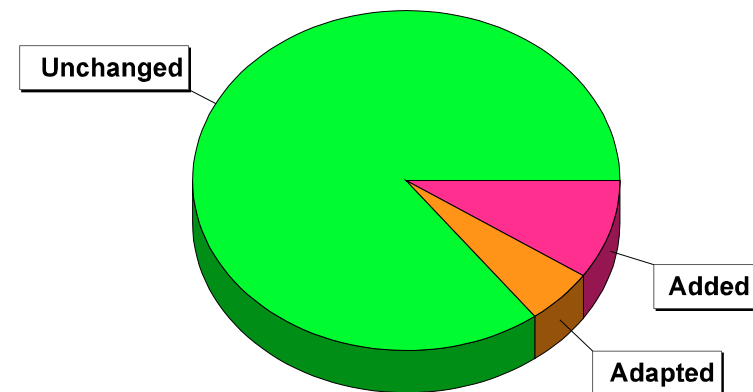
# ***System Metrics***

***Gert Caspersen***  
***gec@terma.com***

## *Data Handling Software Required Adaptations*

	<b>No. of Lines</b>	<b>Ratio</b>
Unchanged	10870	85%
Adapted	687	5%
Added	1189	9%
<b>Total</b>	<b>12746</b>	

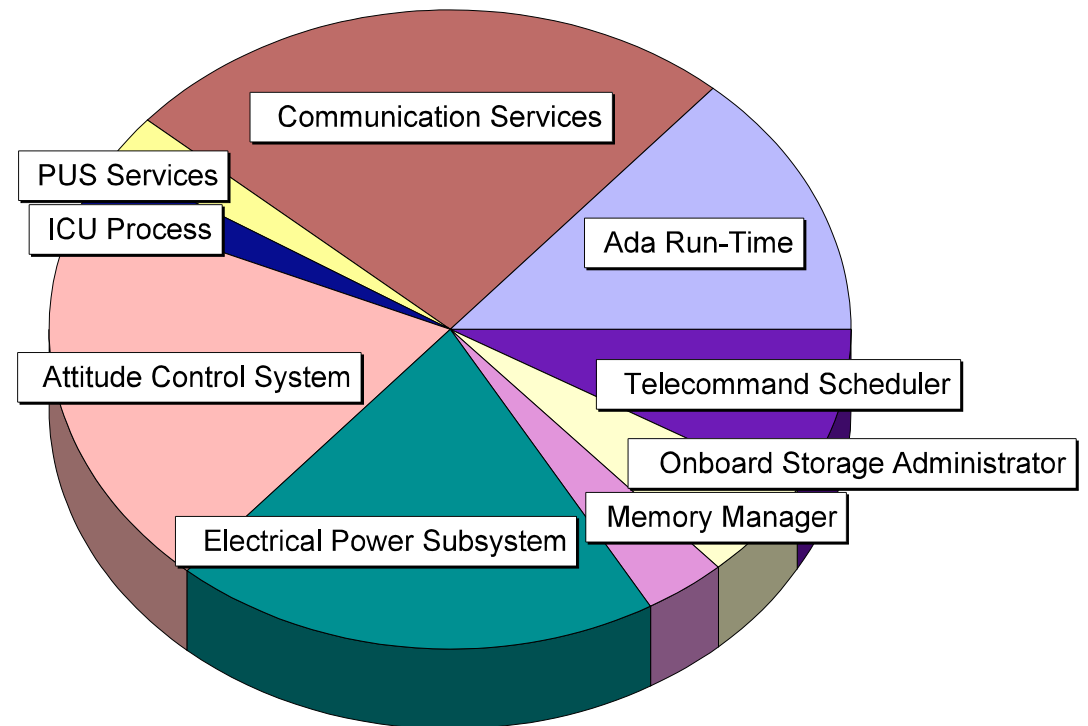
### Demonstration Scenario



## Data Handling Software Code Size

System Components	Size (kbytes)
Ada Run-Time	72
Communication Services	130
PUS Services	13
ICU Process	13
Attitude Control System	105
Electrical Power Subsystem	102
Memory Manager	18
Onboard Storage Administrator	25
Telecommand Scheduler	43
<b>Total</b>	<b>450</b>

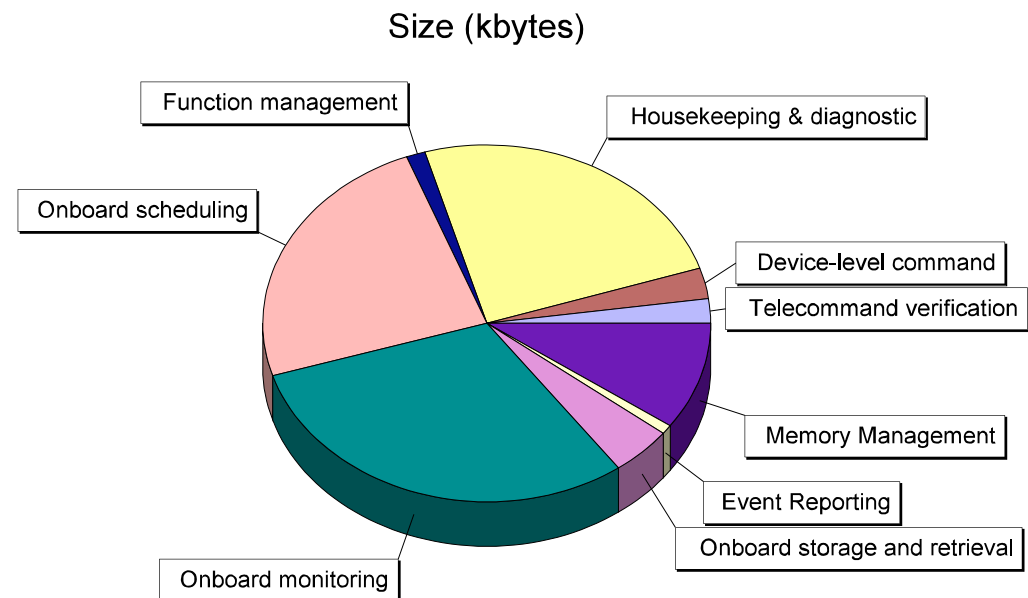
## Code Size

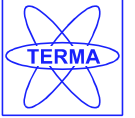


## Data Handling Software Service Code Size

<i>PUS Service</i>	<i>Size (kbytes)</i>
Telecommand verification	4
Device-level command	5
Housekeeping & diagnostic	43
Function management	3
Onboard scheduling	42
Onboard monitoring	54
Onboard storage and retrieval	8
Event reporting	1
Memory management	17
<b>Total</b>	<b>177</b>

### Service Code Size





## *Lessons Learned*

*Gert Caspersen*

*gec@terma.com*

## *Experiences from Adapting OBOSS*

### **Making design & implementation HRT-HOOD compliant**

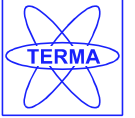
- Few, localised, changes to transform 'hybrid' objects (and the corresponding Ada tasks) into pure cyclic and/or sporadic ones
- Requiring unique task priorities necessitated changes due to non-staticness of generic parameters

### **Porting to ERC32 platform (from 1750A)**

- Changing from a 16- to a 32-bit platform hardly needed to be reflected in the code
- Very few compiler-related changes to Ada code
- Direct use of generics (no 'manual' source code expansion needed)

## *Hard Real-Time HOOD Tool (HRT-HoodNICE)*

- Basic HOOD functionality is mature
- Flexible document generator
  
- The concept of 'shared objects' is very inflexible
- The internal database may get corrupted
- Exporting a project in SIF format and later importing it sometimes results in an error
- When the user violates certain rules, the tool simply crashes with an 'internal error'
- Maintenance quality low (slow reaction, no bug fixes, no usable work-arounds suggested)



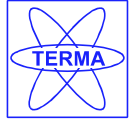
## *Ada Compiler (Aonix)*

- Good language coverage (compared to 1750A compiler)
- Useful utility packages (e.g. Dynamic\_Priorities)
- Code quality not impressive (e.g. index check within attribute-based for-loops)
- Explicitly requested extensive optimisations (by compiler options) often result in failing compilation
- Lacking a 'current exception name' function for error reporting
- The price, particularly for maintenance, is quite high



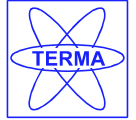
## *Ada Debugger (Aonix)*

- Generally adequate functionality
- Good integration with the ERC32 target simulator
- Cumbersome to set breakpoints in inner generic instances
- Sometimes not possible to 'continue' or 'step' after a breakpoint was hit



## *ERC32 Target Simulator*

- Easily adaptable using Tcl/Tk
- Maintenance provided at no cost
- Cumbersome use of coverage bits
- Price is quite high compared to state-of-the-art simulators for other processors
- Performance is acceptable, but could be improved



## *Lessons Learned*

- Packet Utilisation Standard provides **commonality over missions**
- Reuse only viable if **inter-company reuse is accepted by the onboard software community**
- **10 out of 17** Packet Utilisation Standard services
- It is easier to optimize functionally correct code than it is to make optimized code functionally correct
- **Advantageous** to apply Packet Utilisation Standard to Instrument Control Units
- Reuse approach results in **substantial economic gains**

[http://spd-web.terma.com/Projects/OBOSS/Home\\_Page](http://spd-web.terma.com/Projects/OBOSS/Home_Page)

## *Where to Go?*

- ❓ Is industry prepared to reuse third-party onboard software?
- ❓ Will ground segments, electrical ground support equipment etc. support Packet Utilisation Standard?
- ❓ Will ESA make reuse mandatory to reduce cost?
- ❓ Whatever happened to 'smaller, faster, cheaper'?