ELISA
Enabling Linux in Safety Applications

WORKSHOP

NASA Goddard

F Prime and Linux

Michael Starch – NASA Jet Propulsion Laboratory

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Jet Propulsion Laboratory
California Institute of Technology

Introduction

About the Speaker

- Flight Software Engineer at NASA Jet Propulsion Laboratory
- Community Manager of the F Prime Flight Software Product Line
- Open-Source advocate
- Downlink and Tools lead for the Ingenuity Mars Helicopter

Agenda

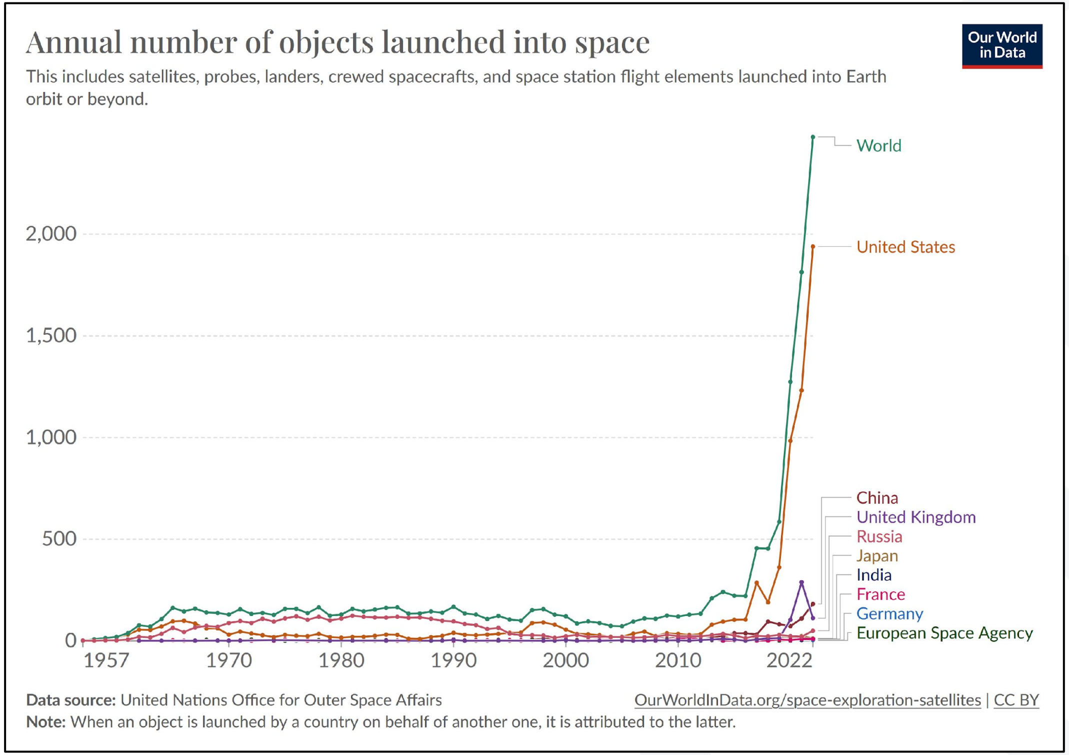
- What is F Prime?
    - A New Paradigm for Space
    - The F Prime Product Line
    - What is F Prime?
    - F Prime Streamlines Flight Software Development
- F Prime and Linux
    - F Prime and Operating Systems
    - F Prime and Linux: Ingenuity Mars Helicopter
    - Ingenuity System Design
- Linux Lessons Learned
    - Linux Supports Efficient Development
    - Plan for Robustness
    - Remember Its Linux!
- Questions

What is F Prime?

ELISA Enabling Linux in Safety Applications

WORKSHOP

A New Paradigm for Space



The F Prime Product Line

The F Prime Software Product Line targets flight missions

- Flight systems
- Particularly well suited for small-scale flight systems, i.e. CubeSats or SmallSats

- Flight instruments and/or sensors requiring software developed for a dedicated processing platform

- Flight “deployables” – freely operating spaceborne flight systems deployed by spacecraft in support of a mission
- Ex. A planetary lander or helicopter

Additionally, F Prime supports

- Simulation and SSE development requiring embedded software

- Software technology development with intended path to flight

F Prime is Open-Source!

What is F Prime?

Software Architecture

- Provides modular components with well-defined interfaces that communicate over ports
- Enables rapid development, portability, high performance, reusability, analyzability, testability

C++ Framework

- Implements the F’ architecture
- Provides basic features such as message queues, threading, and OS abstraction

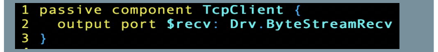
Component Library

- Provides generic components for common capabilities, e.g., command dispatch
- Usable without modification in new software projects

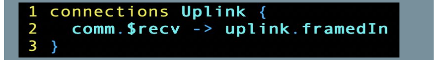
Tool Suite

- Tools for specifying components and connections, automatically generating code
- Tools for FSW unit and integration testing, including a lightweight ground data system

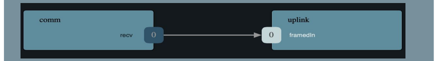
F Prime Streamlines Flight Software Development



```
passive component TcpClient {
  output port $recv: Drv.ByteStreamRecv
}
```

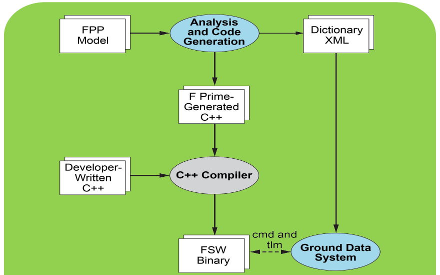


```
connections Uplink {
  comm.$recv -> uplink.framedIn
}
```



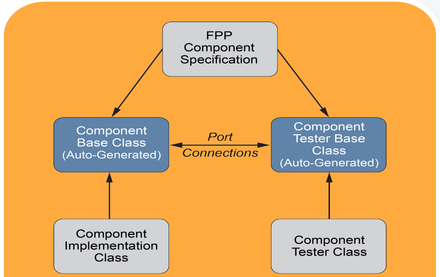
Design

- Developers model F Prime using the F Prime Prime (FPP) domain-specific language
- FPP tools check models for correctness
- Companion tools visualize topology graphs



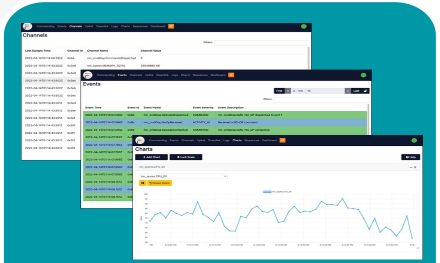
Implementation

- FPP tools translate models to partial implementations in C++
- Developers write mission-specific code
- Developers avoid writing boilerplate


FPP Component Specification
Component Base Class (Auto-Generated)
Port Connections
Component Tester Base Class (Auto-Generated)
Component Implementation Class
Component Tester Class

Unit Testing

- FPP tools generate unit test harnesses from the component models
- Each harness has a mirror-image interface to the component under test



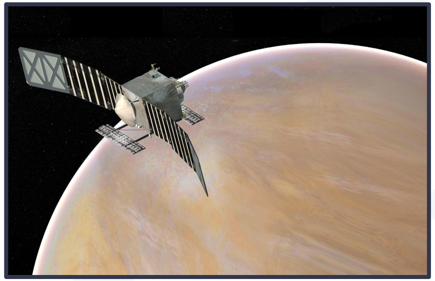
Integration Testing

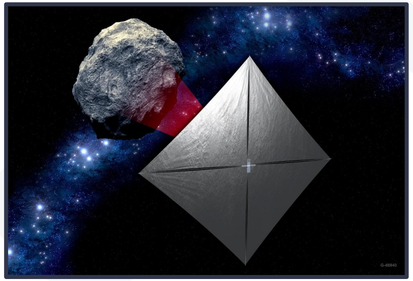
- Integrated Ground Data System supports integration and system testing
  - Uses auto-generated command and telemetry dictionary
  - Provides a Python API for writing automated tests
  - Provides a Web-based GUI
- F Prime can be adapted to support other GDSs

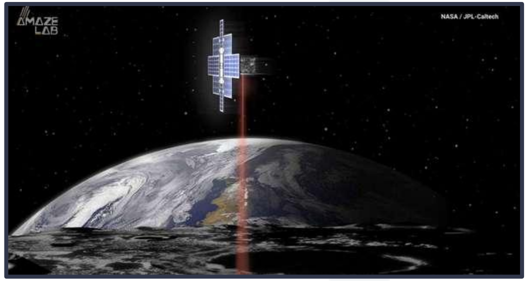
ELISA Enabling Linux in Safety Applications

WORKSHOP

F Prime is Flight Proven!





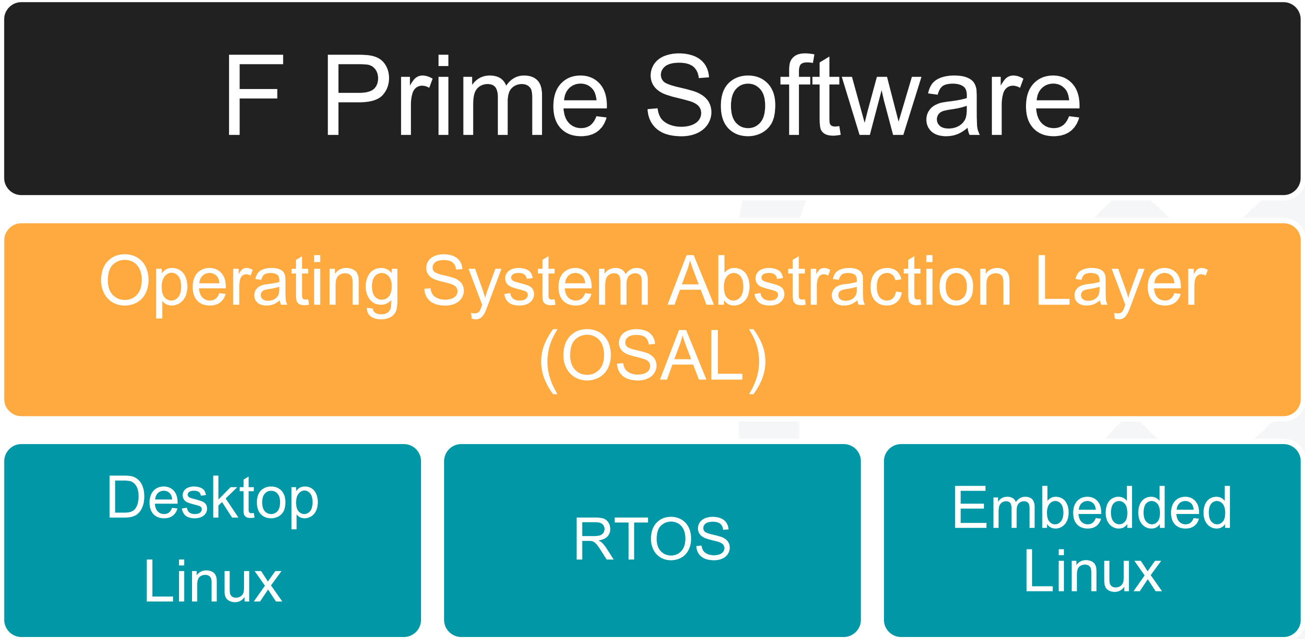


F Prime and Linux

ELISA
Enabling Linux in
Safety Applications

WORKSHOP

F Prime and Operating Systems



F Prime Software

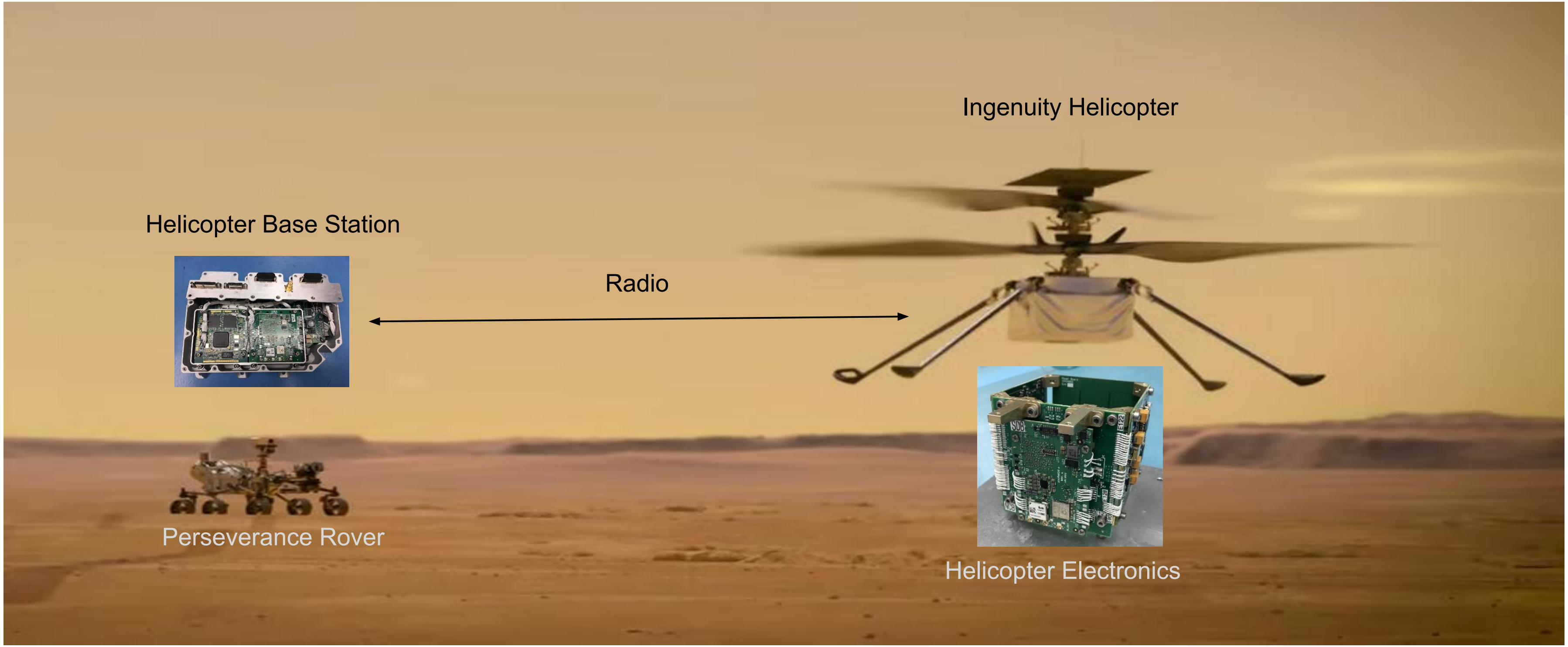
Operating System Abstraction Layer (OSAL)

Desktop Linux

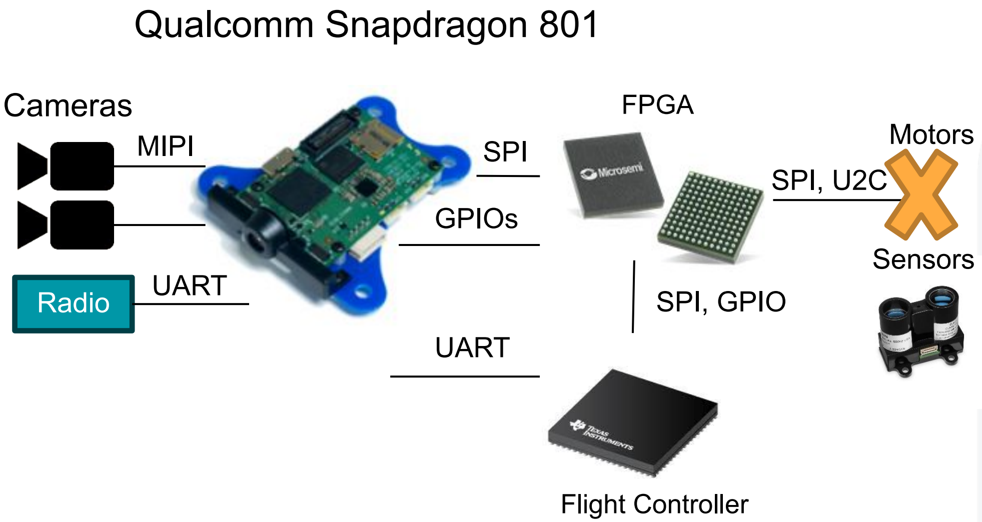
RTOS

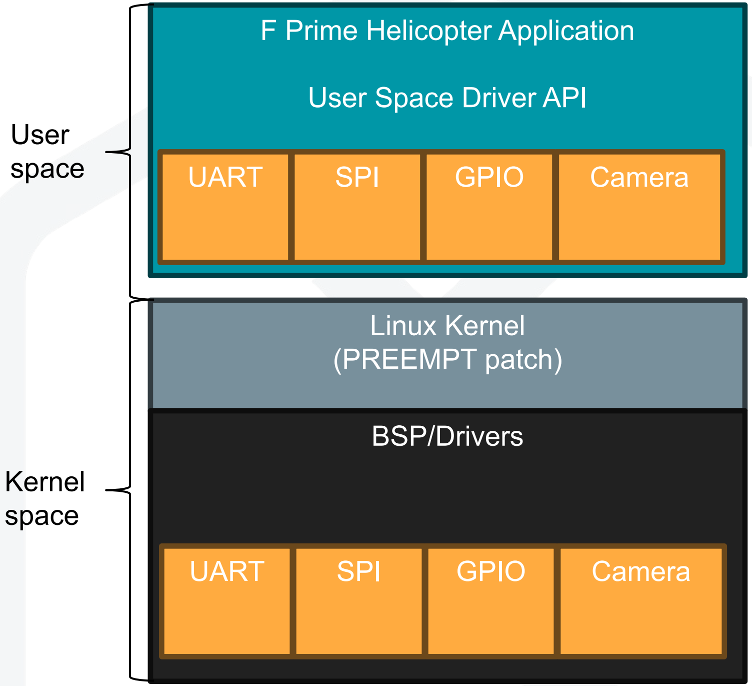
Embedded Linux

F Prime and Linux: Ingenuity Mars Helicopter

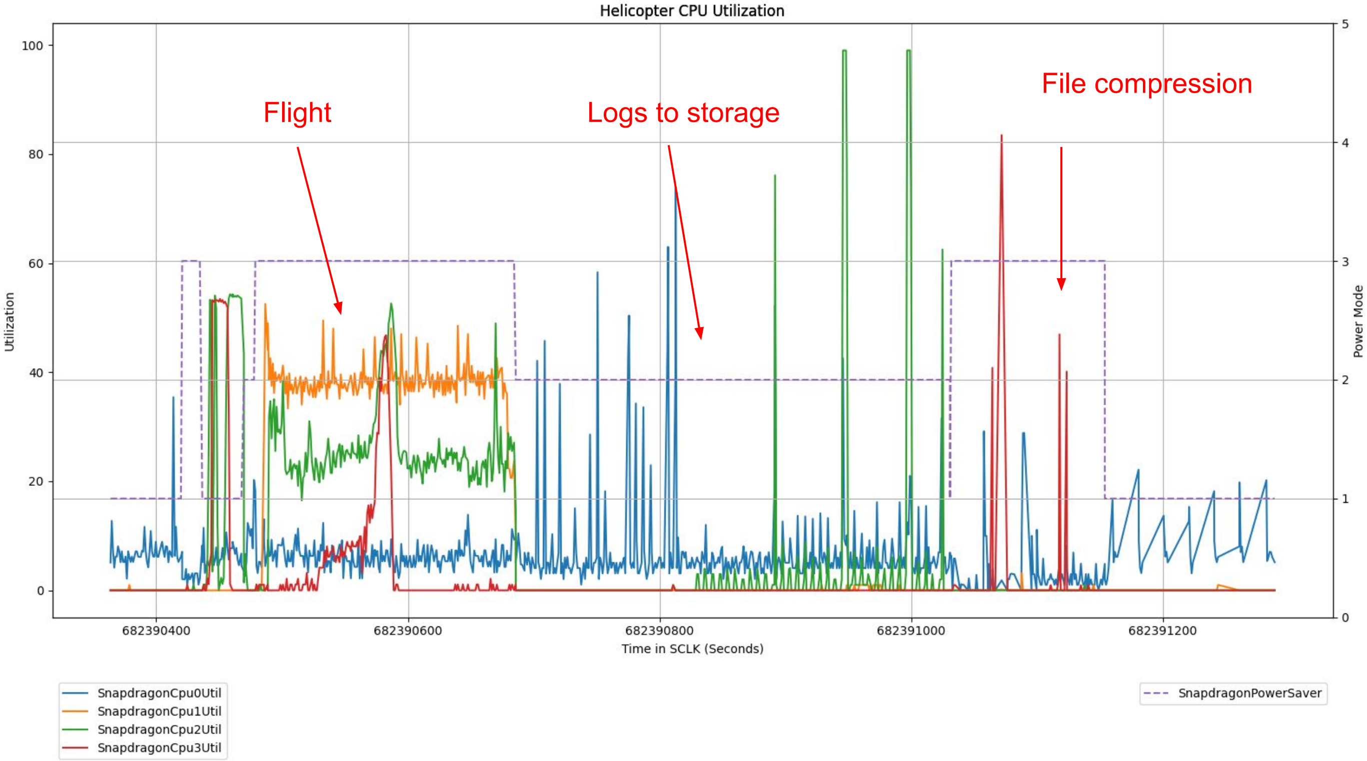


Ingenuity System Design





How Did It Work Out?



Core 0 (Blue)
- Data and logging
- Telecom
- Device I/O

Core 1 (Orange)
- Cameras

Core 2 (Green)
- Visual processing
- Image logging
- Data routing to MCU

Core 3 (Red)
- Guidance/Navigation

Linux Lessons Learned

ELISA
Enabling Linux in
Safety Applications

WORKSHOP

Linux Supports Efficient Development

- Development stations and commodity hardware (Raspberry PIs) can be used to test most code without flat-sat hardware

- Standard user space APIs allow efficient sharing of code between deployments and between projects

- Standard board features like WiFi, USB, Ethernet, etc. made working with the testbed much more efficient

Plan for Robustness

- Not real-time, no RT patch
  - Software designed to handle slips
- File I/O had a performance impact
  - Cache data while flying; persist to disk once on the ground
- Abrupt power-off can cause file corruption
  - Dedicated read-only partition for kernel, golden software images, and launch scripts

Remember Its Linux!

- Shell and Android debugger available for development and testing
- `system()` calls could be used to extend mission functionality
- User space commands are available: bzip2, md5sum, ls, python, rm…

NASA/JPL-Caltech

Questions?

NASA/KSC

Licensing of Workshop Results

All work created during the workshop is licensed under Creative Commons Attribution 4.0 International (CC-BY-4.0) [https://creativecommons.org/licenses/by/4.0/] by default, or under another suitable open-source license, e.g., GPL-2.0 for kernel code contributions.

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.