

# Challenges of MSL Entry, Descent and Landing Validation Or, "7 Years of Terror"

Ann Devereaux, MSL EDL Deputy Lead with  
Rob Manning, MSL Chief Engineer

NASA Workshop of Validation of Autonomous Systems  
August 20, 2012 Caltech



Material presented here was created at the Jet Propulsion Laboratory,  
California Institute of Technology under a contract with the National  
Aeronautics and Space Administration  
(Clearance number CL# 231386)



# But – it wasn't easy

- Overview of Entry, Descent and Landing Implementation
- Independent Autonomous Actors in Play
  - EDL Timeline and Guidance Mode Commander
  - EDL Timeline and Fault Protection
  - Descent Stage Flyaway
- Verification and Validation Approach
- Special Topic: Non-critical actors in play during EDL
- Lessons Learned

# A few of Curiosity's Talents

## An interplanetary spacecraft

- Safely flies 200 million miles
- Targets to within  $\frac{1}{2}$  km at top of Mars atmosphere
- Precision landing to 20x7 km target

## A hypersonic aircraft and lander

- 5.7 km/s  $\rightarrow$  0 in under 6 min
- First autonomously guided Mars entry

## A fault tolerant spacecraft

- Dual string avionics
- Hot swap capability

## An autonomous truck

- Path planning
- Hazard avoidance

## A roving telecomm station

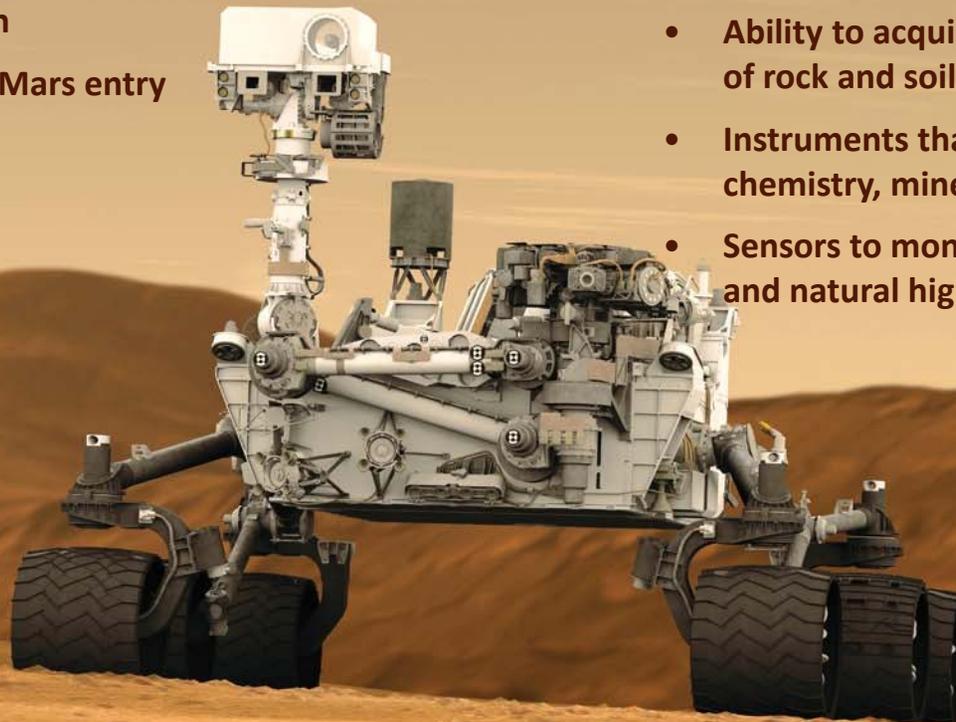
- Dual UHF radios
- X-band links

## A Robotic Field Geologist

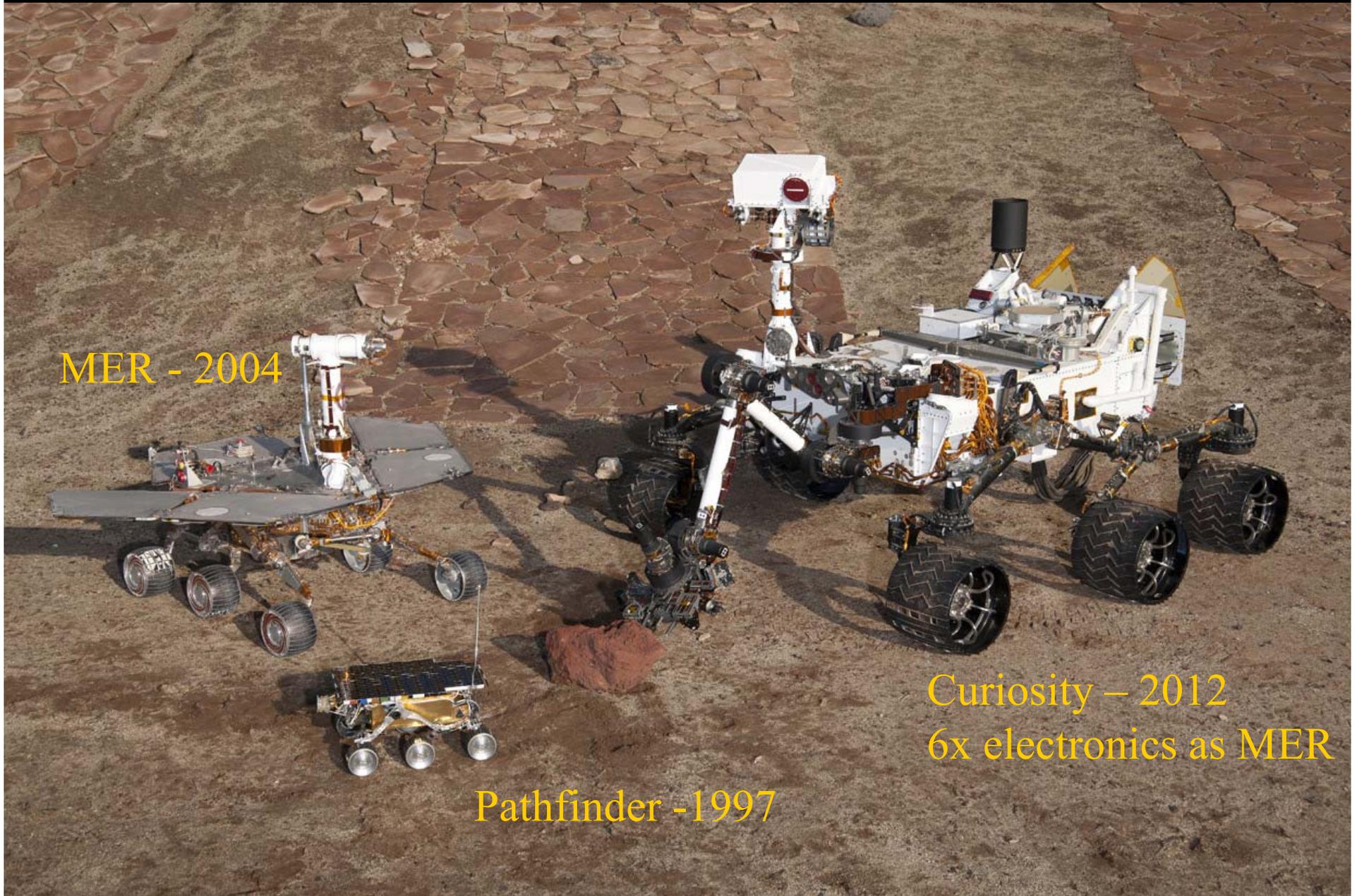
- Long life, ability to traverse many miles over rocky terrain
- Ability to image & survey composition of bedrock and regolith

## A Mobile Geochemical and Environmental Laboratory

- Ability to acquire and process dozens of rock and soil samples
- Instruments that analyze samples for chemistry, mineralogy, and organics
- Sensors to monitor water, weather, and natural high-energy radiation



# Exponential Growth?



MER - 2004

Curiosity - 2012  
6x electronics as MER

Pathfinder - 1997

# MSL Hardware



*Cruise Stage:  
Rover &  
Descent  
stage  
encapsulated,  
with Cruise  
stage flying*

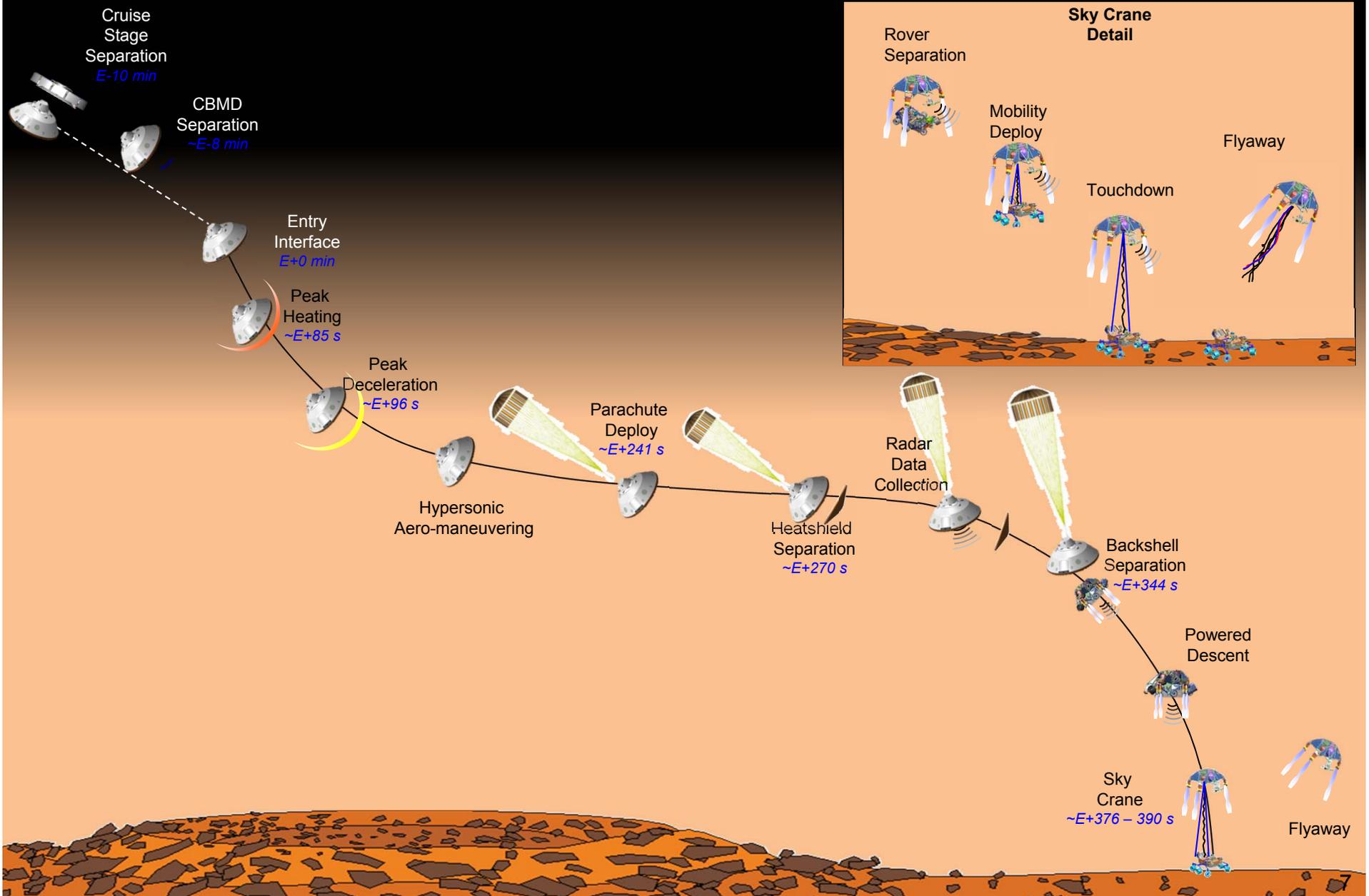


*Descent Stage:  
Lowers Rover to  
surface and then  
flies away*



*Rover: Houses control computer for all stages*

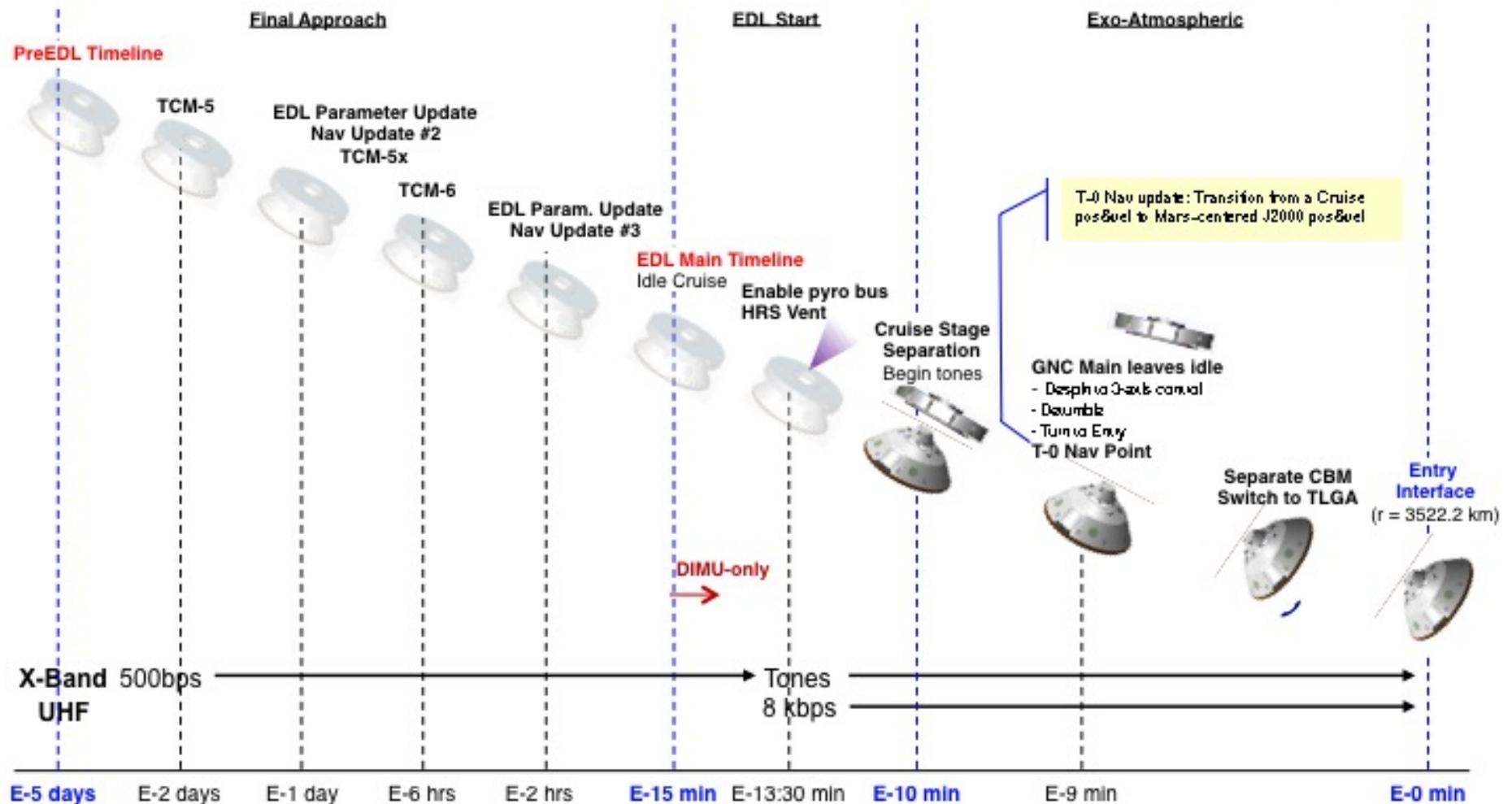
# Entry, Descent and Landing Overview



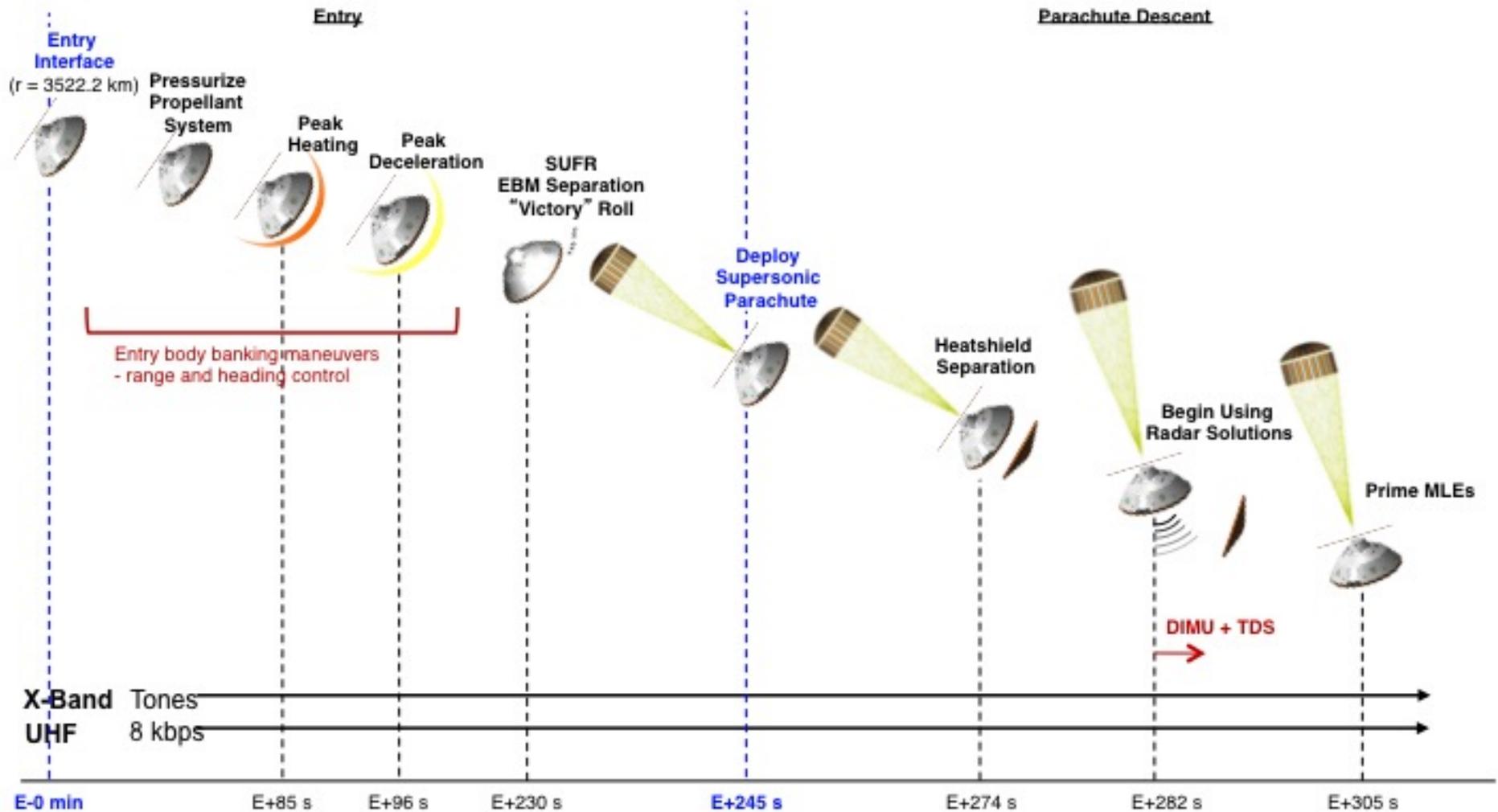
# MSL EDL Design

- Exquisite pas de deux between EDL Timeline actor and GNC Mode Commander actor
- EDL Timeline module
  - Executes sequences of timed events - “Anchors” – set at absolute times (relative to other Anchors) or by GNC triggers (e.g., achieving threshold velocities)
- GNC Mode Commander
  - Focused on flight dynamics modes – entry guidance, flight on parachute, powered flight, landing

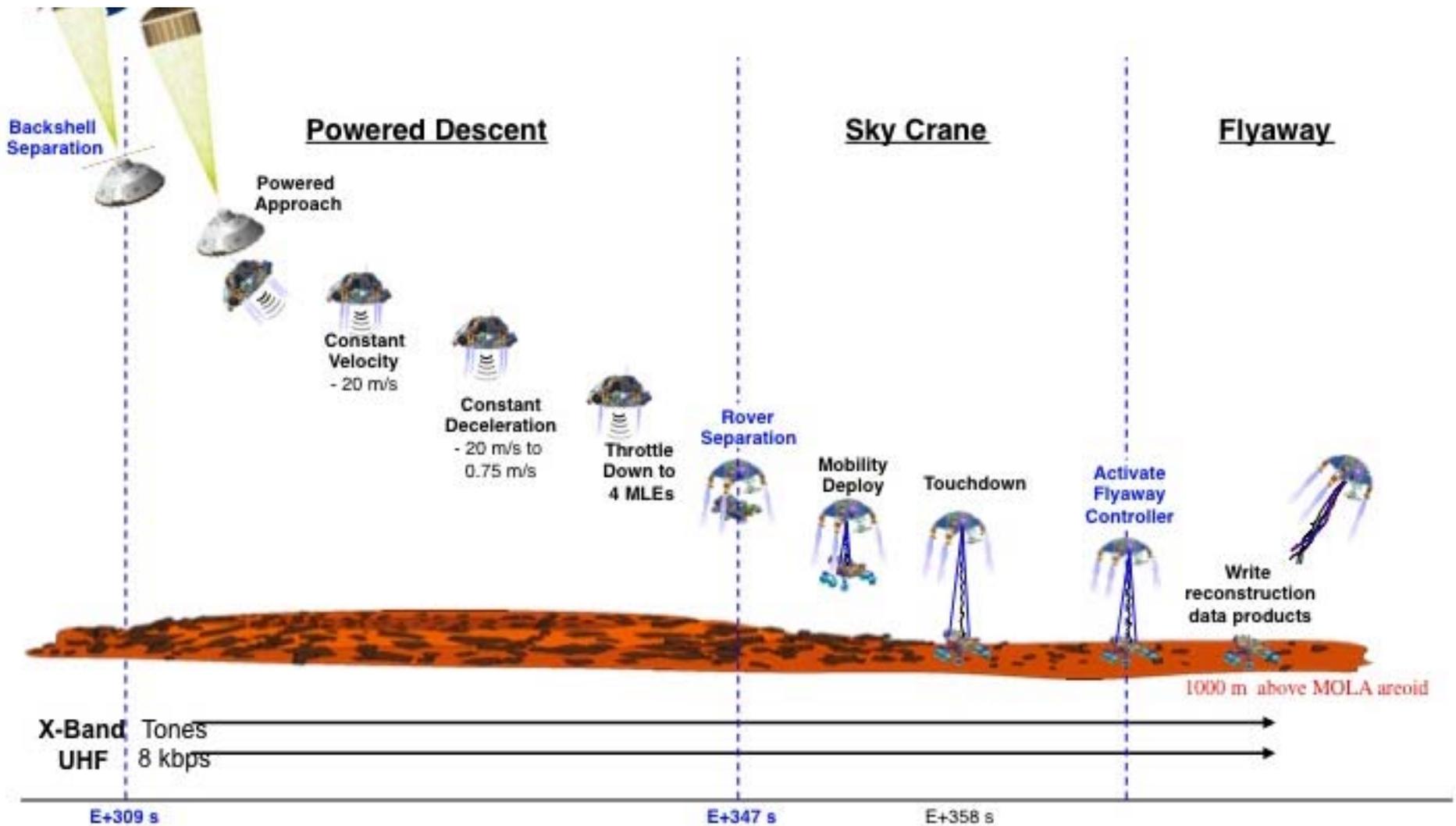
# EDL Timeline – Approach to Entry



# EDL Timeline – Parachute Deploy



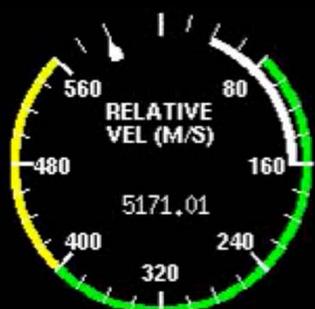
# EDL Timeline – Landing



# 3 Autonomous Control Regimes

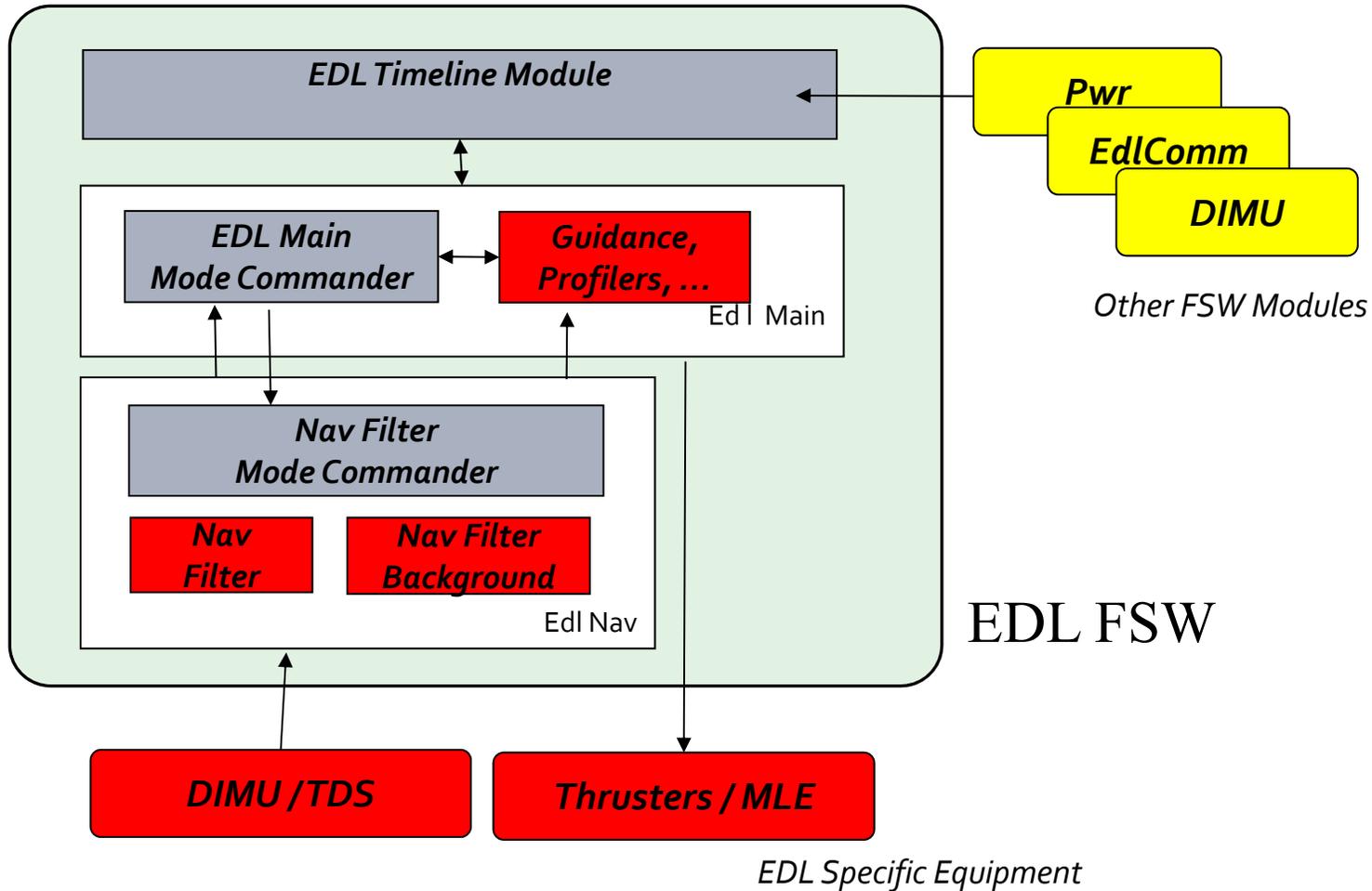
- Entry-5days to Entry-2hrs (last Pre-EDL anchor)
  - EDL Timeline & System Fault Protection (SFP) both running
  - Ground-in-the-loop response to faults
- Entry-2hrs to EDL Main (E-40 min)
  - EDL Timeline & SFP active, ground no longer in play (no commanding)
- EDL Main (E-40 min to landing)
  - EDL Timeline and GNC Mode Commander active
  - SFP deactivated, ground not in loop

# EDL Dashboard Movie

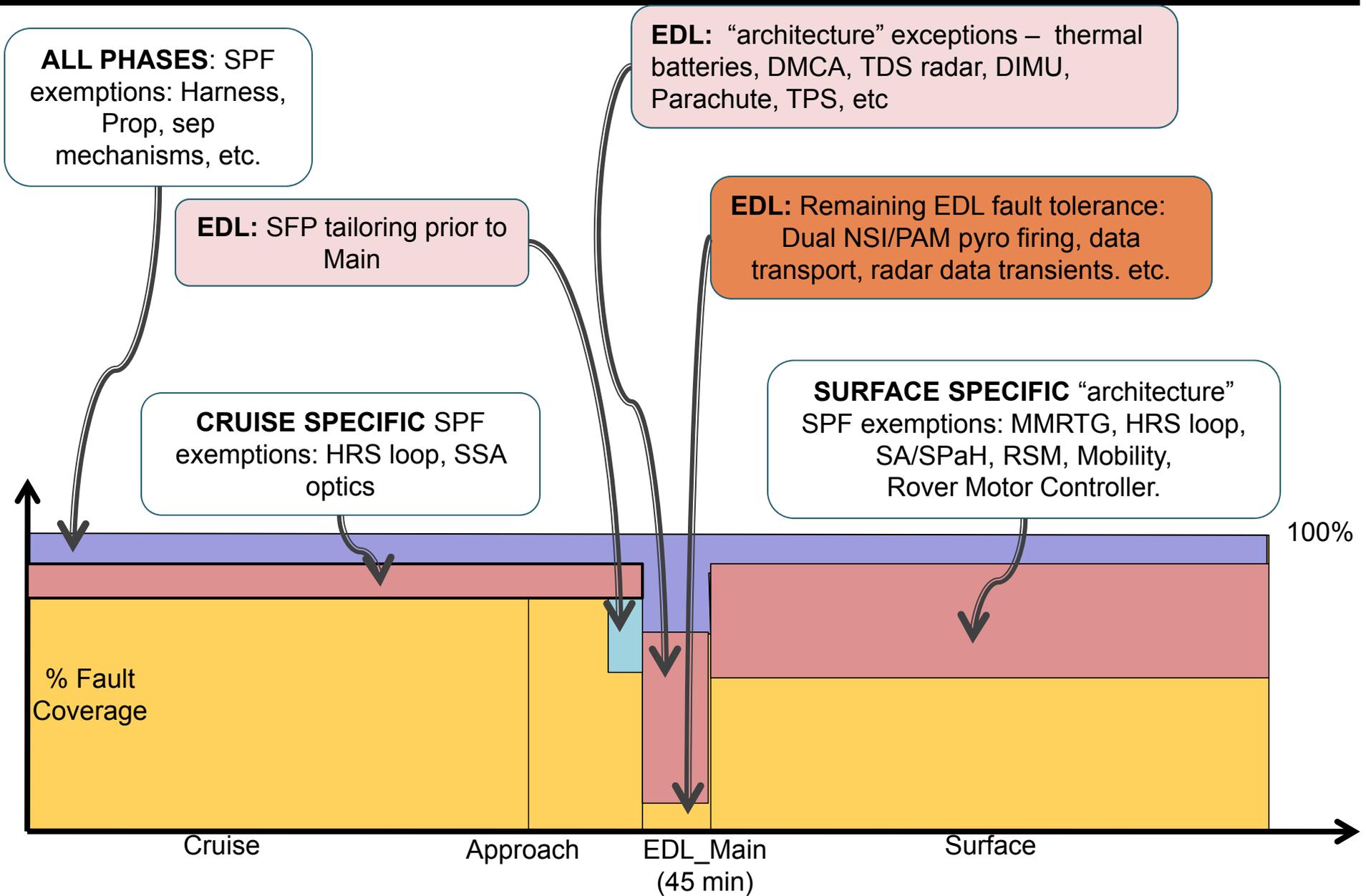


IDLE

# EDL Timeline vs Mode Commander



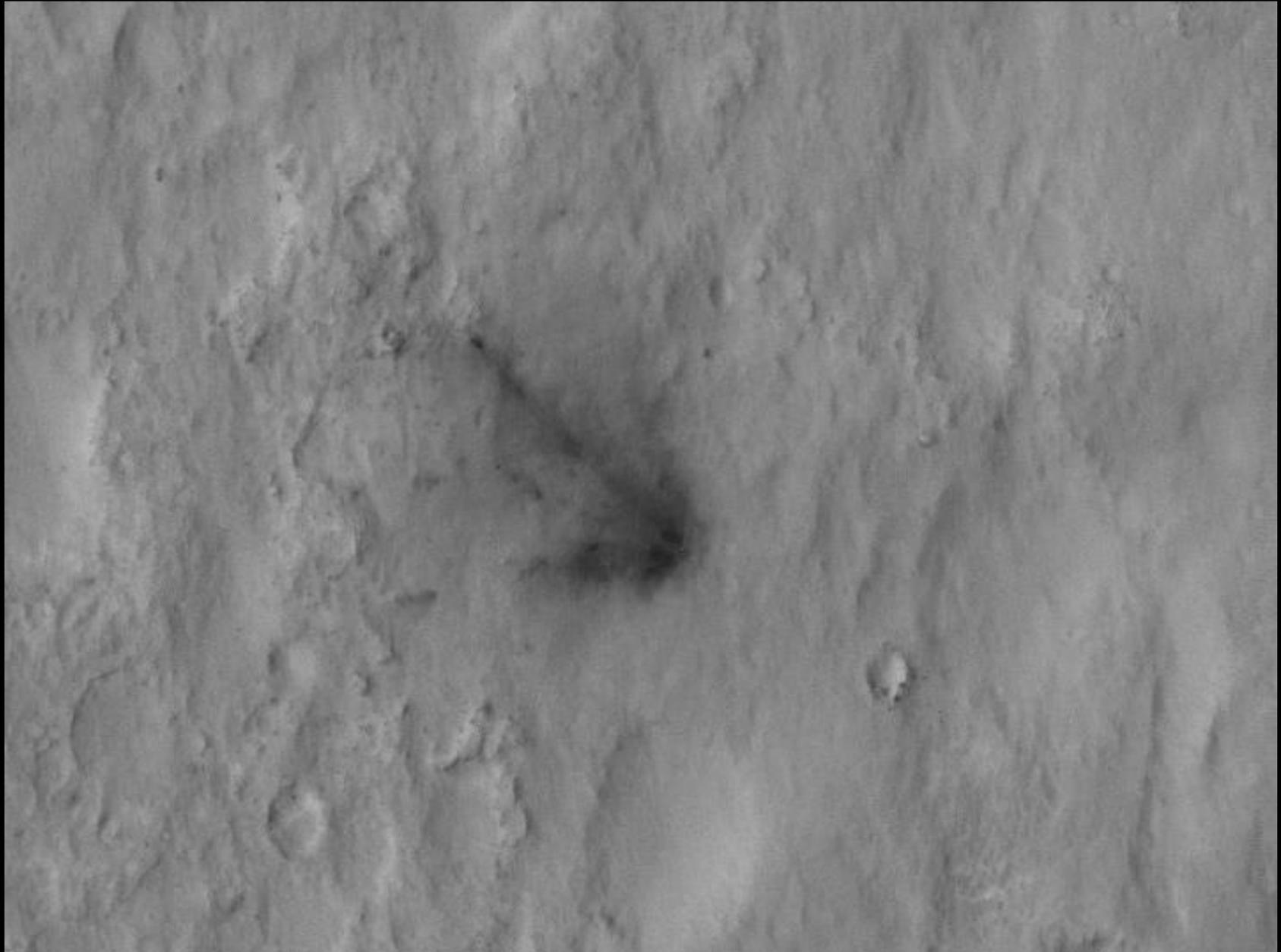
# EDL vs System Fault Protection



# SFP/EDL Risk-Risk Trade

- Allow SFP to remain enabled up through EDL\_Main (beyond window of Ground-in-the-Loop). Cases:
  - True fault that would have killed you
    - SFP response will likely have time to fix it
  - True fault that wouldn't have killed you
    - SFP response kicks off to fix (nice, but unnecessary) but happens to have sensitive part of response overlap window of vulnerability
  - False positive (Transient fault reveals/compounds hard fault in possibly single channel of device)
    - SFP response kicks off and happens to have sensitive part of response overlap window of vulnerability
- Disable SFP after last set of Approach actions (Only EDL autonomous functions after commanding disabled). Cases:
  - True fault that would have killed you
    - SFP response will not run, EDL fails
  - True fault that wouldn't have killed you
    - SFP response will not run, EDL marginally affected but no chance of critically adverse reaction
  - False positive (Transient fault reveals/compounds hard fault in possibly single channel of device)
    - SFP response will not run, no chance of adverse reaction

# Rover vs Descent Stage Flyaway



# Verification/Validation Approach

- MSL's core autonomous systems (e.g. entry descent and landing, fault protection, sleep/wake) assumes that the DESIGN is correct and that any off-nominal event is due to environmental effects or hardware failure.
  - Defects, however few, undermine this assumption.
- Today our primary pathway to eliminate design defects is through systematic testing.
  - One testbed to test cruise and EDL
  - Another testbed to test the rover.
  - Plus some software simulation capability.
- There is not enough time to test all of the permutations and combinations.

# EDL Verification and Validation

## SIMULATION REGIME

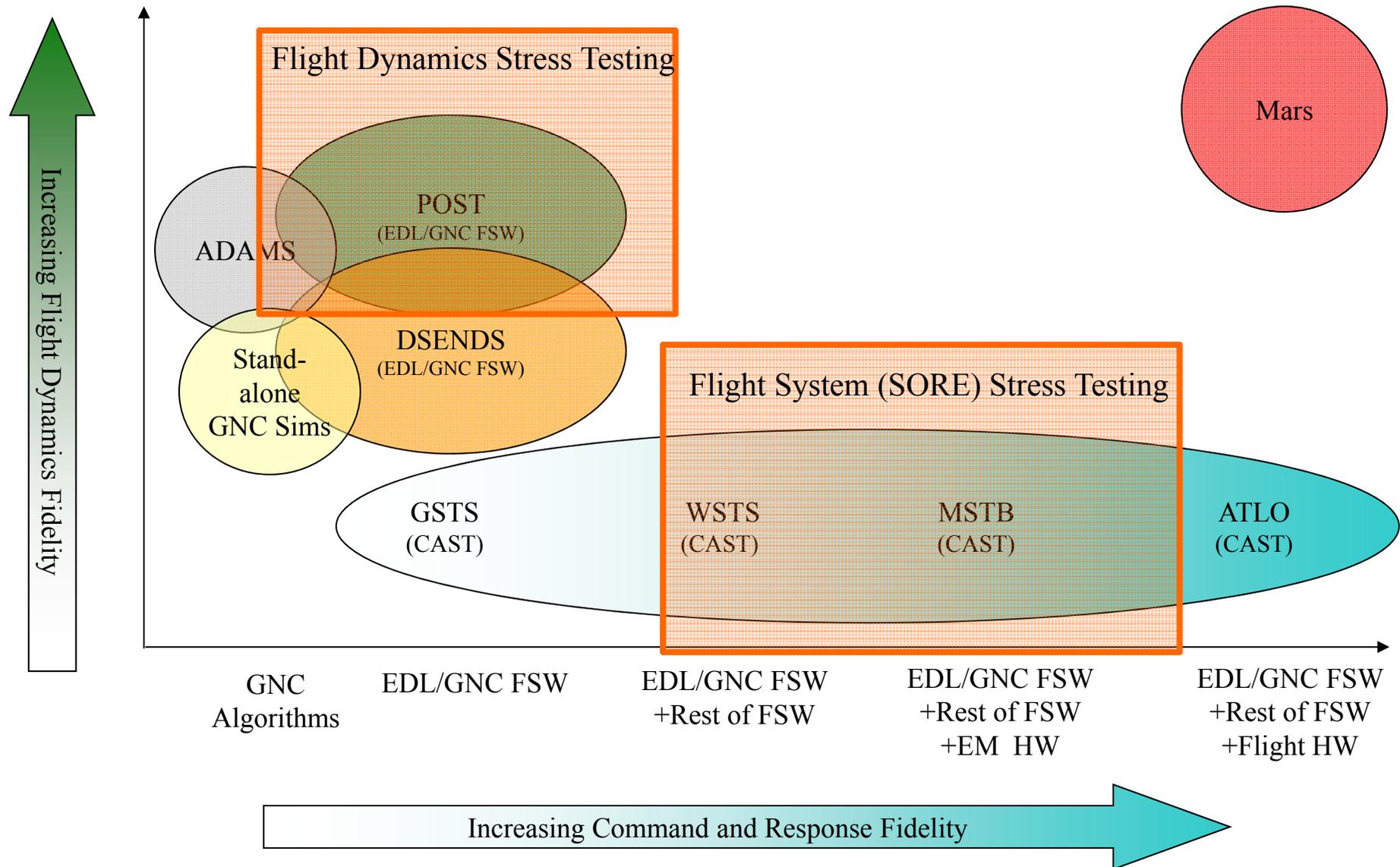
*Flight Dynamics  
(External Behavior/Environmental Interaction)  
End-to-end Simulation  
fed by validated models*

*Hardware (Unit-Software) Verification  
(Unit and Assembly Level Function)  
Subsystem Verification  
Pre-launch testing*

*Command and Response  
(Internal Behavior/HW-SW Interaction)  
Testbeds  
Pre-launch testing*

## FLIGHT SYSTEM AND TESTBED REGIME

# V&V Map



# V&V Summary

- Flight Dynamics
  - Simulation: 100K POST Monte Carlo runs
- Flight System
  - Testbed/Spacebed test: ~ 800 Verification Items
- Stress testing
  - Testbed/Simulation test: ~300 Stress Test cases
- EDL Functional Certifications
  - Testing/Analysis: ~81 individual EFCs containing total ~900 elements of success tree
- “Second Chance” backup FSW testing
  - Testbed/Simulation test: ~300 Verification Items

# How to tell testing is comprehensive?

Consider the ways we can look at the system

- Defined success criteria for landing
  - Pyro timing, computer messaging, dynamics envelopes; criteria all plugged into analysis tools to give green, yellow or red light to each test run
- Address and test Known Knowns
  - Specific Verification Items (pyro functionality, etc) defining proper modes of the Flight System
  - EDL Functional Certifications, defining how the functional components of the system need to behave correctly for overall success
- Address and test Known Unknowns
  - POST Monte Carlo runs, varying atmospheric/flight parameters to bound system performance
  - Fault protection testing, applying known faults to system to verify recovery
- Address and test Unknown Unknowns
  - SORE stress testing, throwing faulted situations at system without defining specific faults that may have caused them (e.g., muting all telemetry)

# Event Tree/EFC Validation

- Requirements-based verification matrix is a necessary but insufficient approach to ensure completeness
  - Not practical to enumerate EDL success with requirements
- MSL EDL utilizes the MER-developed and Phoenix-enhanced “success tree” approach
  - Hierarchical method of enumerating all conditions and events that are required to be successfully executed to ensure EDL success
- Each “node” in the success tree represents a condition or property that must be satisfied and forms the basis for the V&V Matrix (Verification Elements)
- If executed properly, the activities required to satisfy the Verification Elements in the success tree-based V&V Matrix are a superset of those that appear in a pure requirements-based matrix

# EDL Functional Certifications

- Groupings of leaves from EDL success tree are placed into an EFC
- All 892 elements of the success tree placed in 81 EFCs
- EFCs become VI's in DOORS

*Success Tree Elements for EFC #64*

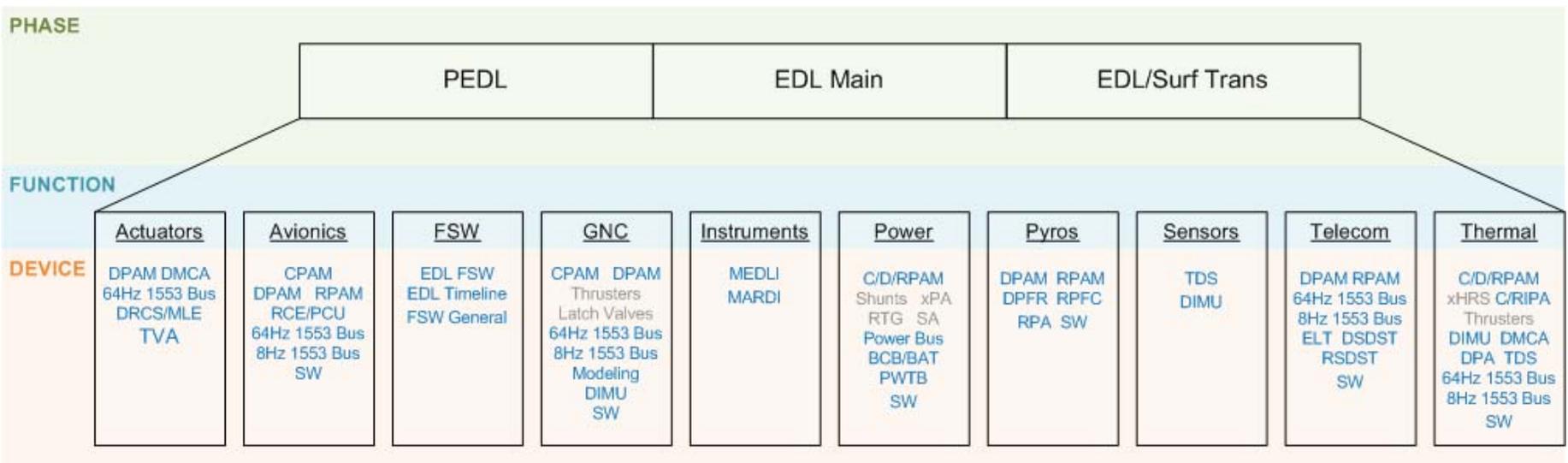
Element	Element Description
8065	Descent Stage Impact
8065.2	Min Safe Flyaway Distance Achieved
8065.4	Min Safe Flyaway Distance Bounded Correctly
8066	DS Impact Response Bounded Correctly
8069	Post-Impact Propellant/Pressurant Tank Failure Behavior Bounded
8070	Tank Depressurization Time Bounded Correctly
8071	Final DS Location Can Be Determined to Expected Accuracy

EFC #	Sample EFCs
41	RCS System Performance and Model Certification
29	Mobility Release/Landing Loads Study
19	Final Aerothermal Review
34	Plume on Parachute Study
64	Descent Stage Impact
17	GNC Alignment Error Budgets Study
13	EDL VAP (all EDL VAP related items)
8	Backshell Separation Trigger Study
39	Propellant Budget Scrub



# Stress Test Definition Process

- Started with “bottom-up” process and collected/brainstormed faults cases with subject matter experts
  - “What scenarios, functions, or actions worry you?”
- Then did “top down” process to map each objective to Phase/Function/Device
  - Met with subject matter experts again to review mapping
    - “Are we stressing each function/device sufficiently? Are there holes in our test coverage?”
    - Some function/device line items ended up not having any test objectives associated with them, and that’s ok!



# Stress Test Validation Regimes

- **Priority 1 –**
  - Faults the system has been specifically designed for and are expected to be survivable
  - Faults that are likely to reveal underlying dependencies
    - Even if they are “extreme” faults that may result in a crash landing
- **Priority 2 –**
  - Faults that may be survivable but have not been explicitly designed for
- **Priority 3 –**
  - Faults that are not expected to be revealing
  - Faults that are not expected to be survivable and we understand the failure mechanism

# MARDI – Do No Harm?



# What ended up being surprises?

- Actual EDL \*much cleaner\* than any test we'd done
  - Many tests compromised by faulty sim/support equipment or test operator error
  - Actual EDL environments were much more benign than simulated environments
  - Most feared problems were “boogiemen”: undefined noise causing resets, etc., which did not materialize
- Conclusions – real EDL did not stress our system, and by extension, our testing program

# Lessons Learned

- Define success criteria early
  - Formal success criteria not defined until months before launch, making test analysis a laborious process
- Automate testing early
  - Many tests blown because of bad set up, operator error, equipment problems
  - Non-repeatable tests make overall readiness story problematic
- Limit independent autonomous actors in design
  - Activity space blows up – often fixes were made specifically to LIMIT amount of testing required, not to achieve best design per se
- Did we spend the right amount of time on the right things?
  - A lot of test time spent on off-nominals, because they were scary. But EDL ended up completely nominal – might have been better to do a lot of testing with nominal cases, just to ensure they were grooved in?