

Ranger: 3D Space Mission Visualization in a Web Browser

Marc Pomerantz

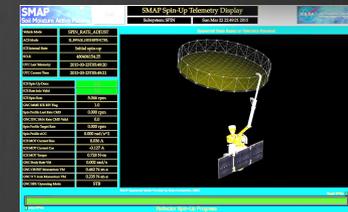
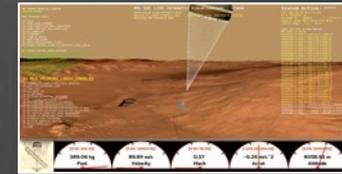
Technical Group Supervisor
Information and Data Management Group (393G)
Mission Control Systems Section
Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, CA, 91109

Contact: Marc.I.Pomerantz@jpl.nasa.gov

- ❖ Engineering Goals
- ❖ Past Visualization Projects
- ❖ Customer Domains
- ❖ 3D visualization in prehistoric and modern times
- ❖ *Ranger* Overview
- ❖ *Ranger* System Architecture
- ❖ Who is Using Ranger?
- ❖ *Flow* Server Architecture
- ❖ Summary

Build an easily re-useable and data driven visualization system to help mission engineers understand the state of their system and that is *engineering accurate* and can be reconfigured to support a variety of space and robotic mission scenarios

- We have supported a variety of NASA/JPL flight and engineering projects by providing real-time telemetry visualization and situational awareness displays, vehicle commanding, modeling and simulation
 - Hypercube missile defense simulation/visualization
 - Autonomous Feature and Star Tracking simulation/visualization
 - Synthetic Aperture Radar instrument coverage
 - **Galileo** telemetry display of Huygens probe health and status
 - Support of various robotic vehicle operations, simulation and proposal tasks
 - **MSL** landing night telemetry visualization
 - **LDSD** Flight Director display (health and status of vehicle). Used during I&T and operations during two test flights.
 - Hundreds of channels displayed and splashdown predict model in the loop
 - **M2020 Sampling Arm** telemetry display and robot commanding software used during system hardware and software development
 - **SMAP** boom deployment and antenna spin-up telemetry displays for operations
 - Multi-agent autonomy tasks. 3D situational awareness displays.
 - **M2020 EDL** testbed telemetry visualization (like our MSL viz). In development
 - Mission visualization for JPL formulation. In development



Interactive 3D Visualization software required high-end workstations and expensive graphics cards

SMAP
Soil Moisture Active Passive

Vehicle Mode: SPIN_RATE_ADJUST

ACS Mode: 3L_RWA31_HIGHSPINCTRL

ICE Internal State: initial spin-up

SCLK: 48006154.25

UTC Last Telemetry: 2015-03-23T05:49:20

UTC Current Time: 2015-03-23T05:49:21

ICE Spin Up Done: 0.0

ICE Rate Info Valid: 0.0

ICE Spin Rate: 5.046 rpm

CNC MMB ICE BSV Flag: 1.0

Spin Profile Last Rate CMD: 0.000 rpm

CNC EXC(SMA) Rate CMD Valid: 0.0

Spin Profile Target Rate: 0.000 rpm

Spin Profile ACC: 0.000 rad/s^2

ICE MOT Current Size: 0.036 A

ICE MOT Current Cos: -0.127 A

ICE MOT Torque: 0.728 N-m

ICE Body Rate VM: 0.002 rad/s

CNC OB EST Momentum VM: 0.462 N-m-s

CNC X Y Axis Momentum VM: 0.235 N-m-s

CNC SRU Operating Mode: STB

SMAP Spin-Up Telemetry Display

Subsystem: SPIN Sun Mar 22 22:49:21 2015 6290

Telemetry Beards Received

NASA Jet Propulsion Laboratory
California Institute of Technology

Spacecraft State Based on Telemetry Received

SMAP Spacecraft Model Provided by Brian Kumanochi (188)

Target RPMs

Initial RPMs

Reflector Spin-Up Progress

SMAP
Antenna
Deployment

MC State Display Legend

Current State

Not Received State

10 OPEN_LOOP_START

11 OPEN_LOOP_GET_ALE_THROTTLES

12 OPEN_LOOP_GET_ALE_THROTTLES

13 OPEN_LOOP_GET_ALE_THROTTLES

14 OPEN_LOOP_GET_ALE_THROTTLES

15 OPEN_LOOP_GET_ALE_THROTTLES

16 OPEN_LOOP_GET_ALE_THROTTLES

17 WAIT_FOR_ENTRY

18 WAIT_FOR_ENTRY

19 WAIT_FOR_ENTRY

20 TDS_NAV_INIT

21 WAIT_FOR_ENTRY

22 WAIT_FOR_ENTRY

23 WAIT_FOR_ENTRY

24 WAIT_FOR_ENTRY

25 WAIT_FOR_ENTRY

26 WAIT_FOR_ENTRY

27 WAIT_FOR_ENTRY

28 WAIT_FOR_ENTRY

29 WAIT_FOR_ENTRY

30 WAIT_FOR_ENTRY

31 WAIT_FOR_ENTRY

32 WAIT_FOR_ENTRY

33 WAIT_FOR_ENTRY

34 WAIT_FOR_ENTRY

35 WAIT_FOR_ENTRY

36 WAIT_FOR_ENTRY

37 WAIT_FOR_ENTRY

38 WAIT_FOR_ENTRY

39 WAIT_FOR_ENTRY

40 WAIT_FOR_ENTRY

41 WAIT_FOR_ENTRY

42 WAIT_FOR_ENTRY

43 WAIT_FOR_ENTRY

44 WAIT_FOR_ENTRY

45 WAIT_FOR_ENTRY

46 WAIT_FOR_ENTRY

47 WAIT_FOR_ENTRY

48 WAIT_FOR_ENTRY

49 WAIT_FOR_ENTRY

50 WAIT_FOR_ENTRY

51 WAIT_FOR_ENTRY

52 WAIT_FOR_ENTRY

53 WAIT_FOR_ENTRY

54 WAIT_FOR_ENTRY

55 WAIT_FOR_ENTRY

56 WAIT_FOR_ENTRY

57 WAIT_FOR_ENTRY

58 WAIT_FOR_ENTRY

59 WAIT_FOR_ENTRY

60 WAIT_FOR_ENTRY

61 WAIT_FOR_ENTRY

62 WAIT_FOR_ENTRY

63 WAIT_FOR_ENTRY

64 WAIT_FOR_ENTRY

65 WAIT_FOR_ENTRY

66 WAIT_FOR_ENTRY

67 WAIT_FOR_ENTRY

68 WAIT_FOR_ENTRY

69 WAIT_FOR_ENTRY

70 WAIT_FOR_ENTRY

71 WAIT_FOR_ENTRY

72 WAIT_FOR_ENTRY

73 WAIT_FOR_ENTRY

74 WAIT_FOR_ENTRY

75 WAIT_FOR_ENTRY

76 WAIT_FOR_ENTRY

77 WAIT_FOR_ENTRY

78 WAIT_FOR_ENTRY

79 WAIT_FOR_ENTRY

80 WAIT_FOR_ENTRY

81 WAIT_FOR_ENTRY

82 WAIT_FOR_ENTRY

83 WAIT_FOR_ENTRY

84 WAIT_FOR_ENTRY

85 WAIT_FOR_ENTRY

86 WAIT_FOR_ENTRY

87 WAIT_FOR_ENTRY

88 WAIT_FOR_ENTRY

89 WAIT_FOR_ENTRY

90 WAIT_FOR_ENTRY

91 WAIT_FOR_ENTRY

92 WAIT_FOR_ENTRY

93 WAIT_FOR_ENTRY

94 WAIT_FOR_ENTRY

95 WAIT_FOR_ENTRY

96 WAIT_FOR_ENTRY

97 WAIT_FOR_ENTRY

98 WAIT_FOR_ENTRY

99 WAIT_FOR_ENTRY

100 WAIT_FOR_ENTRY

MSL EDL LIVE Telemetry Visualization - Chase

Receiving MC State Telemetry Data

Receiving J2000 Telemetry Data

Receiving J2000 Telemetry Data

Last recd: N/A

System Active: *****

UTC: 2015 03 22 22:49:21.370

SCLK: 48006154.25

Altitude (m): 9144.59

Velocity (m/s): 113.14

Latitude (deg): 1.12

Longitude (deg): 112.8

Latitude (deg): 1.12

Longitude (deg): 112.8

Last S/C Alt & Pos Recvd

397502201.17 5544.4 (-1.289511, 4.288417, 0.170204)

397502202.17 5511.4 (-1.289522, 4.288417, 0.170204)

397502203.17 5477.4 (-1.289533, 4.288417, 0.170204)

397502204.17 5443.4 (-1.289544, 4.288417, 0.170204)

397502205.17 5409.4 (-1.289555, 4.288417, 0.170204)

397502206.17 5375.4 (-1.289566, 4.288417, 0.170204)

397502207.17 5341.4 (-1.289577, 4.288417, 0.170204)

397502208.17 5307.4 (-1.289588, 4.288417, 0.170204)

397502209.17 5273.4 (-1.289599, 4.288417, 0.170204)

397502210.17 5239.4 (-1.289610, 4.288417, 0.170204)

397502211.17 5205.4 (-1.289621, 4.288417, 0.170204)

397502212.17 5171.4 (-1.289632, 4.288417, 0.170204)

397502213.17 5137.4 (-1.289643, 4.288417, 0.170204)

397502214.17 5103.4 (-1.289654, 4.288417, 0.170204)

397502215.17 5069.4 (-1.289665, 4.288417, 0.170204)

397502216.17 5035.4 (-1.289676, 4.288417, 0.170204)

397502217.17 5001.4 (-1.289687, 4.288417, 0.170204)

397502218.17 4967.4 (-1.289698, 4.288417, 0.170204)

397502219.17 4933.4 (-1.289709, 4.288417, 0.170204)

397502220.17 4899.4 (-1.289720, 4.288417, 0.170204)

397502221.17 4865.4 (-1.289731, 4.288417, 0.170204)

397502222.17 4831.4 (-1.289742, 4.288417, 0.170204)

397502223.17 4797.4 (-1.289753, 4.288417, 0.170204)

397502224.17 4763.4 (-1.289764, 4.288417, 0.170204)

397502225.17 4729.4 (-1.289775, 4.288417, 0.170204)

397502226.17 4695.4 (-1.289786, 4.288417, 0.170204)

397502227.17 4661.4 (-1.289797, 4.288417, 0.170204)

397502228.17 4627.4 (-1.289808, 4.288417, 0.170204)

397502229.17 4593.4 (-1.289819, 4.288417, 0.170204)

397502230.17 4559.4 (-1.289830, 4.288417, 0.170204)

397502231.17 4525.4 (-1.289841, 4.288417, 0.170204)

397502232.17 4491.4 (-1.289852, 4.288417, 0.170204)

397502233.17 4457.4 (-1.289863, 4.288417, 0.170204)

397502234.17 4423.4 (-1.289874, 4.288417, 0.170204)

397502235.17 4389.4 (-1.289885, 4.288417, 0.170204)

397502236.17 4355.4 (-1.289896, 4.288417, 0.170204)

397502237.17 4321.4 (-1.289907, 4.288417, 0.170204)

397502238.17 4287.4 (-1.289918, 4.288417, 0.170204)

397502239.17 4253.4 (-1.289929, 4.288417, 0.170204)

397502240.17 4219.4 (-1.289940, 4.288417, 0.170204)

397502241.17 4185.4 (-1.289951, 4.288417, 0.170204)

397502242.17 4151.4 (-1.289962, 4.288417, 0.170204)

397502243.17 4117.4 (-1.289973, 4.288417, 0.170204)

397502244.17 4083.4 (-1.289984, 4.288417, 0.170204)

397502245.17 4049.4 (-1.289995, 4.288417, 0.170204)

397502246.17 4015.4 (-1.290006, 4.288417, 0.170204)

397502247.17 3981.4 (-1.290017, 4.288417, 0.170204)

397502248.17 3947.4 (-1.290028, 4.288417, 0.170204)

397502249.17 3913.4 (-1.290039, 4.288417, 0.170204)

397502250.17 3879.4 (-1.290050, 4.288417, 0.170204)

397502251.17 3845.4 (-1.290061, 4.288417, 0.170204)

397502252.17 3811.4 (-1.290072, 4.288417, 0.170204)

397502253.17 3777.4 (-1.290083, 4.288417, 0.170204)

397502254.17 3743.4 (-1.290094, 4.288417, 0.170204)

397502255.17 3709.4 (-1.290105, 4.288417, 0.170204)

397502256.17 3675.4 (-1.290116, 4.288417, 0.170204)

397502257.17 3641.4 (-1.290127, 4.288417, 0.170204)

397502258.17 3607.4 (-1.290138, 4.288417, 0.170204)

397502259.17 3573.4 (-1.290149, 4.288417, 0.170204)

397502260.17 3539.4 (-1.290160, 4.288417, 0.170204)

397502261.17 3505.4 (-1.290171, 4.288417, 0.170204)

397502262.17 3471.4 (-1.290182, 4.288417, 0.170204)

397502263.17 3437.4 (-1.290193, 4.288417, 0.170204)

397502264.17 3403.4 (-1.290204, 4.288417, 0.170204)

397502265.17 3369.4 (-1.290215, 4.288417, 0.170204)

397502266.17 3335.4 (-1.290226, 4.288417, 0.170204)

397502267.17 3301.4 (-1.290237, 4.288417, 0.170204)

397502268.17 3267.4 (-1.290248, 4.288417, 0.170204)

397502269.17 3233.4 (-1.290259, 4.288417, 0.170204)

397502270.17 3199.4 (-1.290270, 4.288417, 0.170204)

397502271.17 3165.4 (-1.290281, 4.288417, 0.170204)

397502272.17 3131.4 (-1.290292, 4.288417, 0.170204)

397502273.17 3097.4 (-1.290303, 4.288417, 0.170204)

397502274.17 3063.4 (-1.290314, 4.288417, 0.170204)

397502275.17 3029.4 (-1.290325, 4.288417, 0.170204)

397502276.17 2995.4 (-1.290336, 4.288417, 0.170204)

397502277.17 2961.4 (-1.290347, 4.288417, 0.170204)

397502278.17 2927.4 (-1.290358, 4.288417, 0.170204)

397502279.17 2893.4 (-1.290369, 4.288417, 0.170204)

397502280.17 2859.4 (-1.290380, 4.288417, 0.170204)

397502281.17 2825.4 (-1.290391, 4.288417, 0.170204)

397502282.17 2791.4 (-1.290402, 4.288417, 0.170204)

397502283.17 2757.4 (-1.290413, 4.288417, 0.170204)

397502284.17 2723.4 (-1.290424, 4.288417, 0.170204)

397502285.17 2689.4 (-1.290435, 4.288417, 0.170204)

397502286.17 2655.4 (-1.290446, 4.288417, 0.170204)

397502287.17 2621.4 (-1.290457, 4.288417, 0.170204)

397502288.17 2587.4 (-1.290468, 4.288417, 0.170204)

397502289.17 2553.4 (-1.290479, 4.288417, 0.170204)

397502290.17 2519.4 (-1.290490, 4.288417, 0.170204)

397502291.17 2485.4 (-1.290501, 4.288417, 0.170204)

397502292.17 2451.4 (-1.290512, 4.288417, 0.170204)

397502293.17 2417.4 (-1.290523, 4.288417, 0.170204)

397502294.17 2383.4 (-1.290534, 4.288417, 0.170204)

397502295.17 2349.4 (-1.290545, 4.288417, 0.170204)

397502296.17 2315.4 (-1.290556, 4.288417, 0.170204)

397502297.17 2281.4 (-1.290567, 4.288417, 0.170204)

397502298.17 2247.4 (-1.290578, 4.288417, 0.170204)

397502299.17 2213.4 (-1.290589, 4.288417, 0.170204)

397502300.17 2179.4 (-1.290600, 4.288417, 0.170204)

389.06 kg Fuel

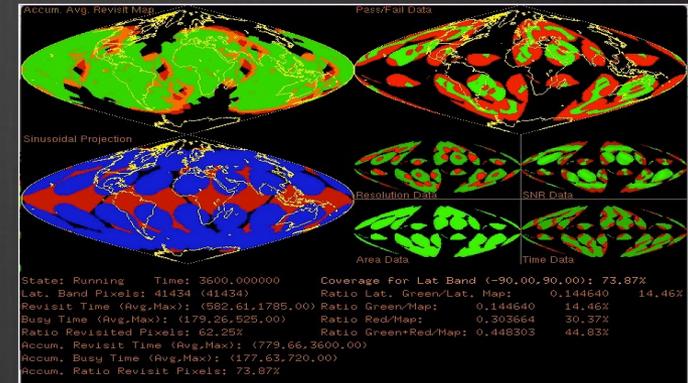
113.14 m/s Velocity

0.47 Mach

-2.32 m/s^2 Accel

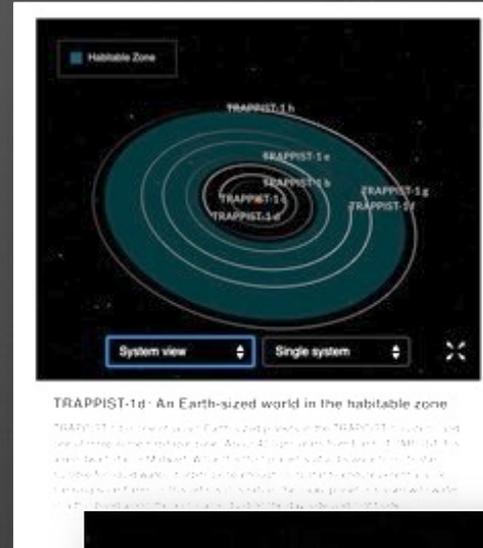
9144.59 m Altitude

MSL Landing
Night



Synthetic
Aperture
Radar
coverage

Interactive 3D Visualization
 software can run inside a web
 browser and on typical laptop
 and mobile devices



NASA EXOPLANET EXPLORATION
 Planets Beyond Our Solar System

TRAPPIST-1 b
 A potentially rocky world, larger than Earth

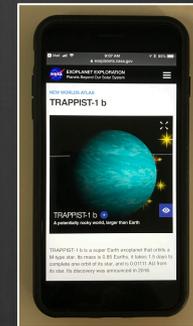
TRAPPIST-1 b is a super Earth exoplanet that orbits a M-type star. Its mass is 0.85 Earths, it takes 1.5 days to complete one orbit of its star, and is 0.01111 AU from its star. Its discovery was announced in 2016.

PLANET TYPE	DISCOVERY DATE
Super Earth	2016

MASS	PLANET RADIUS
0.85 Earths	1.086 x Earth

ORBITAL RADIUS	ORBITAL PERIOD
0.01111 AU	1.5 days

[Back to list](#)



Customer Domains

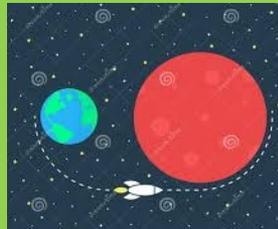
Flight Mission Testing
and Operations



Education and Public
Outreach



Mission Formulation
and Proposals



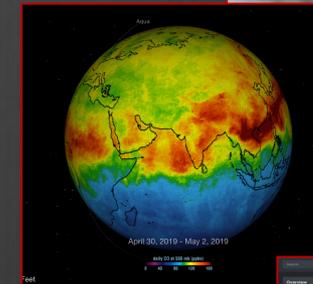
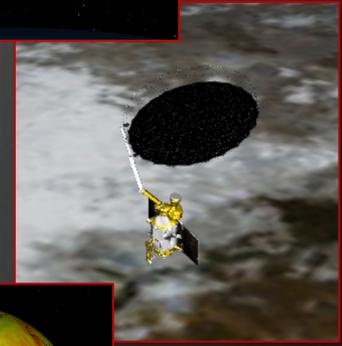
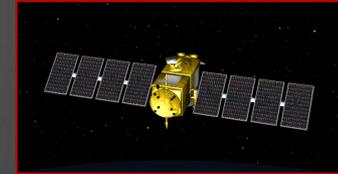
Robotic System
Development and
Testing



Ranger Overview

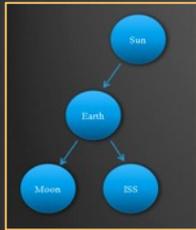
Engineering Accurate Visualization

- Ranger is designed to *accurately and correctly* visualize a 3D representation of spacecraft and robotic vehicle missions
 - Use of supporting mission data and documentation for applications (CAD, GNC Ref. Frames, environment info, imagery, etc.)
 - All calculations double precision on the CPU
 - Frame centric system (no world coordinate system). Preserves numerical accuracy
 - Automated build and regression testing for Ranger code using Jenkins
- Visualizations runs locally on device graphics hardware in a web browser and on desktop and mobile devices
 - GPU acceleration and programming via shaders
- Can be driven from multiple data sources
 - Physics-based models, hardware/software testbeds, mission design tools, telemetry streams, ephemeris data
 - Science data or instrument coverage overlays (static or animated)
- Maintain data security by running server and browser on a secure network
 - Authorization at the web server level
 - Authentication at the network level
- *Flow* is Ranger's backend server that processes data to be shipped to clients
- Immersive VR support (HTC Vive)
- UI support using ReactJS or D3 for integrated dashboard widgets, controls, analytic visuals
- Over 90 Wiki pages of documentation on [Github.jpl.nasa.gov](https://github.com/jpl.nasa.gov), including examples and tutorials



Why is Ranger Multi-Mission?

Mission Specific Input Data



Scene Description
(json)



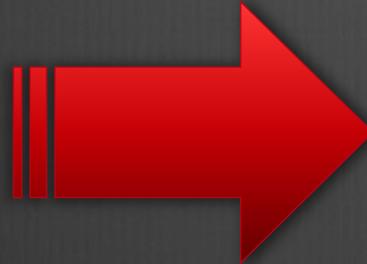
Visual Assets
(CAD, imagery)



User Interface
Assets
(HTML, CSS)

Data drives Ranger

The same application
code can provide a
completely different
user experience



by changing visual
assets, vehicle data,
ephemerides or user
interface

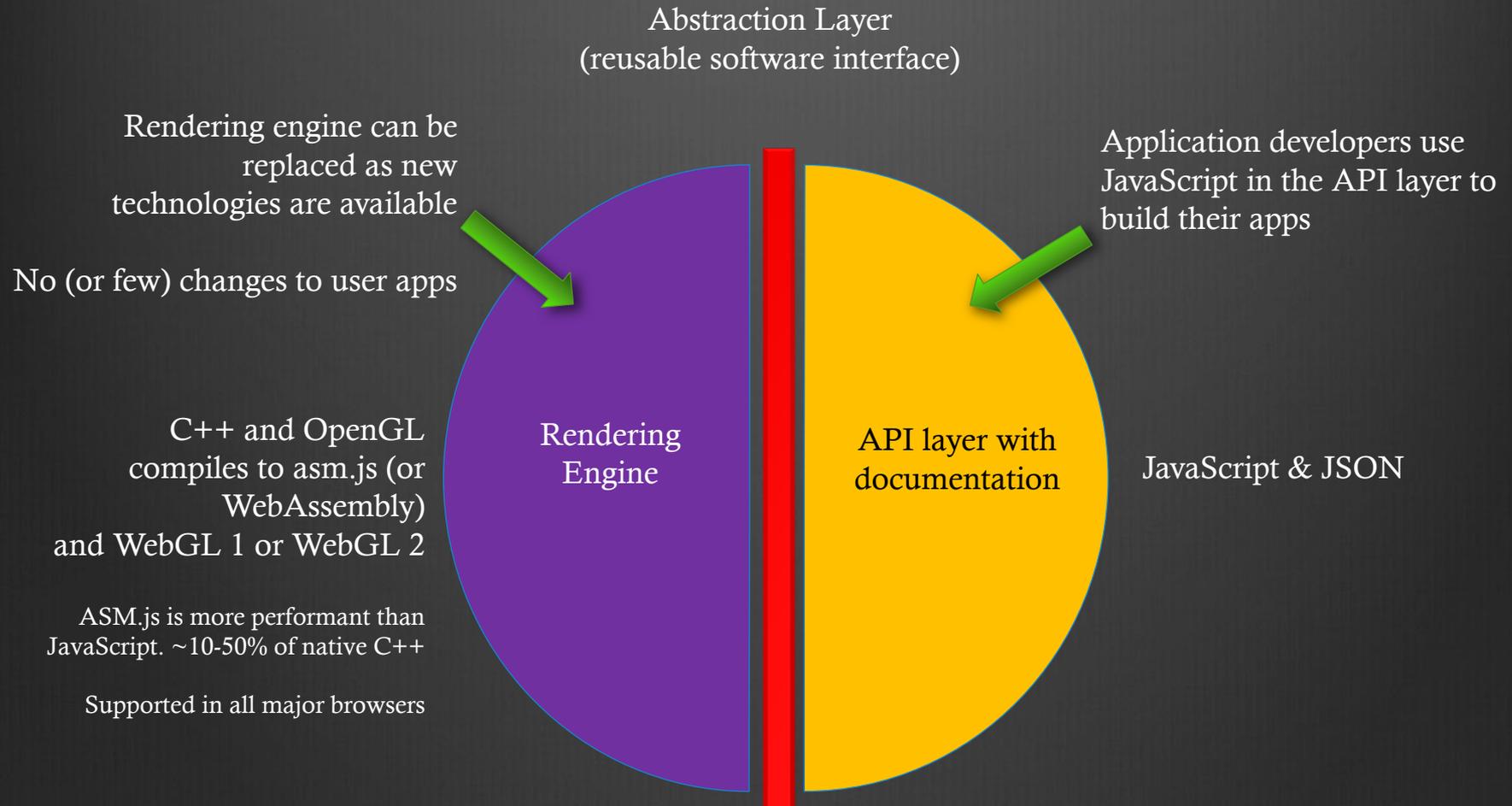
Ranger Core changes infrequently and is shared across applications. Application specific layer can have reusable elements but typically changes from app to app

Ranger Core

C++/OpenGL compiled
using Emscripten, to
ASM.js and WebGL
(faster execution speed
then native JavaScript)

Multi-mission
Application Specific
User Interface

Ranger Engine Design



Ranger has been designed with an abstraction layer between the JavaScript Application Programmers Interface (API) and the rendering engine that makes it possible to replace the renderer as new technologies arise, with minimal changes to the API layer or documentation



Design and Fabrication

Integration and Test

Launch

Operations

Multi-dimensional view into the flight system's health, status and localization.
 Predict models in the loop

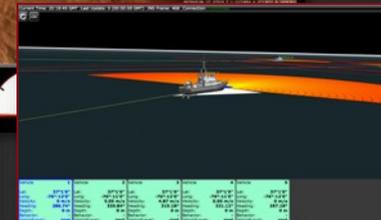
Formulation

Help develop innovative, cost-effective, mission concepts that satisfy mission requirements and communicate designs to engineers and stakeholders

("In The Loop" visualization during trade studies)

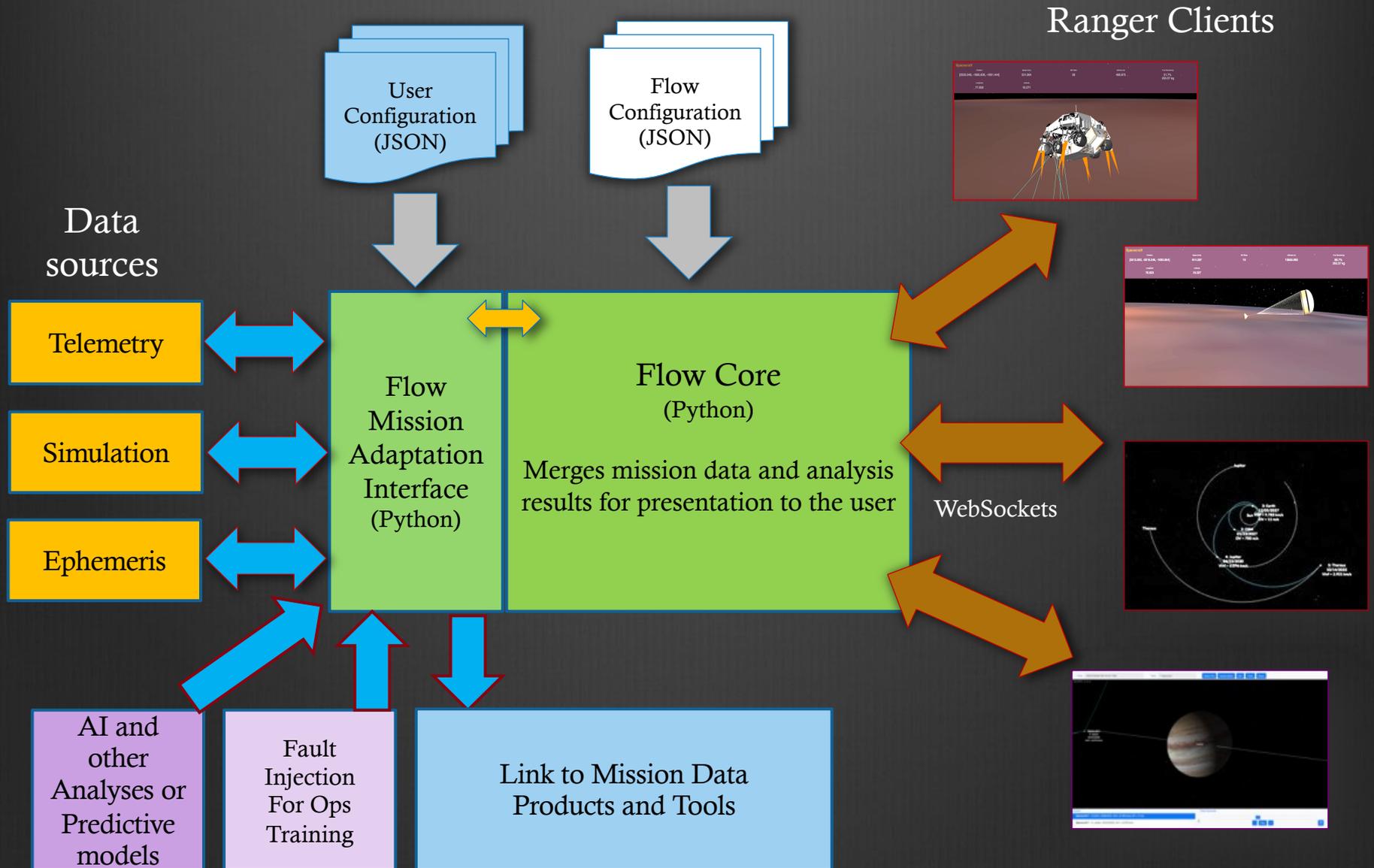
Future Use:

Virtual Operations for Training
 Train new ops personnel without requiring access to actual spacecraft or vehicles

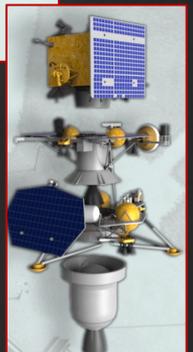
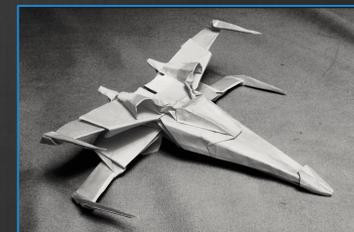


Typical Ranger System Architecture

Multi-mission interface for Mission Information Analysis, Introspection and Control



- Recent and currently supported projects:
 - Engineering Projects
 - 2017-2019 JPL Formulation Application (SCOPE) (JPL Section 312)
 - 2017-2019 JPL Swarms of autonomous boats (JPL Section 347)
 - 2019 M2020 EDL visualization
 - JPL Education and Public Outreach
 - FY16 Exoplanet - <https://exoplanets.nasa.gov/alien-worlds/strange-new-worlds/>
 - FY17 Eclipse - <https://eyes.jpl.nasa.gov/eyes-on-eclipse-web-app.html>
 - FY18 Eyes on Exoplanets - <https://exoplanets.nasa.gov/eyes-on-exoplanets/>
 - FY19 Earth-Now - <https://climate.nasa.gov/earth-now/>
 - Not only for visualizations of physical systems and situational awareness
 - Visualize cyber-physical/virtual systems
 - Exploded view
 - Unfold the origami spacecraft to display internals
 - Display alarm states in context of the flight system
 - Flight rule violations
 - Future?
 - Europa Clipper avionics displays
 - Psyche
 - Machine Learning integrated w/ visualization for mission ops



Flow

Ranger's Telemetry-to-Visualization Interface

Provide an easy to configure, multi-mission,
data interface to drive Ranger visualizations

- Support multiple simultaneous visualization application clients
- Support data coming from multiple, simultaneous data sources
- Support various data types (telemetry, simulation, ephemeris, alarm states, etc.)
- Support mission-provided or third-party analysis (predictive models, AI, etc.)
- Shorten development time for data source developers by providing an easy to use Python API
- Exception handling, logging, connect/re-connect for clients
- Connect to existing, non-Python data sources (Matlab, Simulation, Ops tools, telemetry, etc.) using typical Unix inter-process communication (sockets, pipes, shared memory, pub/sub messaging, etc.)

- Ranger applications have been deployed for multiple customers since 2016
 - Visualizations can be driven by a variety of data sources (telemetry, ephemeris, simulation)
 - Applications can be embedded in existing web pages
- Mission configurations use the Ranger core software and require:
 - Mission specific CAD and environment models
 - Mappings between telemetry items and changes to Ranger app visuals
 - Mission specific UI/UX as needed. UI elements, graphs/plots, ReactJS/D3
- Not only for visualizations of physical systems and situational awareness
 - Visualize cyber-physical/virtual systems.
 - Unfold the origami spacecraft to display internals and data
 - Display telemetry alarm states in context of the flight system or subsystem
 - Flight rule violations
 - Train operations staff by using playback data with injected anomalies
- Predictive models in the loop with mission telemetry to compare predicts vs. actuals

Backups

- C++11 / Javascript / HTML5
 - CMake and Emscripten build C++ into a JavaScript module
 - Webpack assembled Javascript assets into a bundled web app
- C++ to Javascript and ASM.js (or WebAssembly) via Emscripten
 - ASM.js performance from C++ code is 10-50% of native code performance
 - WebAssembly advertised performance is near native code performance
 - Now supported in all major browsers
 - low-level assembly-like language with a JavaScript API
- JSON Scene description
 - Describing the scene as data allows many scenes to be built with zero JavaScript knowledge
- JSON Scene management API
 - Allow interacting with the engine from anywhere using a web socket
- Continuous Integration use Jenkins
- Art Pipeline – Blender to Ranger
- Regression Test suite

- C++ responsible for core renderer and heavy computation
 - OpenGL ES 2.0 Renderer
 - GLSL shaders. i.e. Lighting or atmosphere effects
 - Ephemerides calculation
 - Computational utilities
 - *Note:* Supports future linking with C/C++-based modules such as NAIF Spice Library, predictive models or science data processing into Ranger
- JavaScript library supports the scene API
 - Main event loop
 - User interaction
 - Camera processing
 - Scene graph evaluation
- HTML5 provides user interaction layer

- Emscripten is a Javascript backend for the LLVM compiler
- C, fortran, C++, ruby, rust, and many more languages are available as LLVM front ends
- Emscripten can compile to ASM.js or WebAssembly
- ASM.js performance from C++ code is 10-50% of native code performance
- Performance is substantially better than pure JavaScript for heavy computations