Cal State University, Los Angeles

The SAID Testbed

Friday, December 4th 2020

Welcome Agenda - Matthew

- SRS document - Jerome
- Runaway task - Nicholas
- Memory leak - Alex Huang
- Denial of Service - Vivian
- Invalid Command Sequence - Aaron
- Single bit error - Alexander Lopez
- Project Management Review
  - Samantha
- Ethics - Joshua

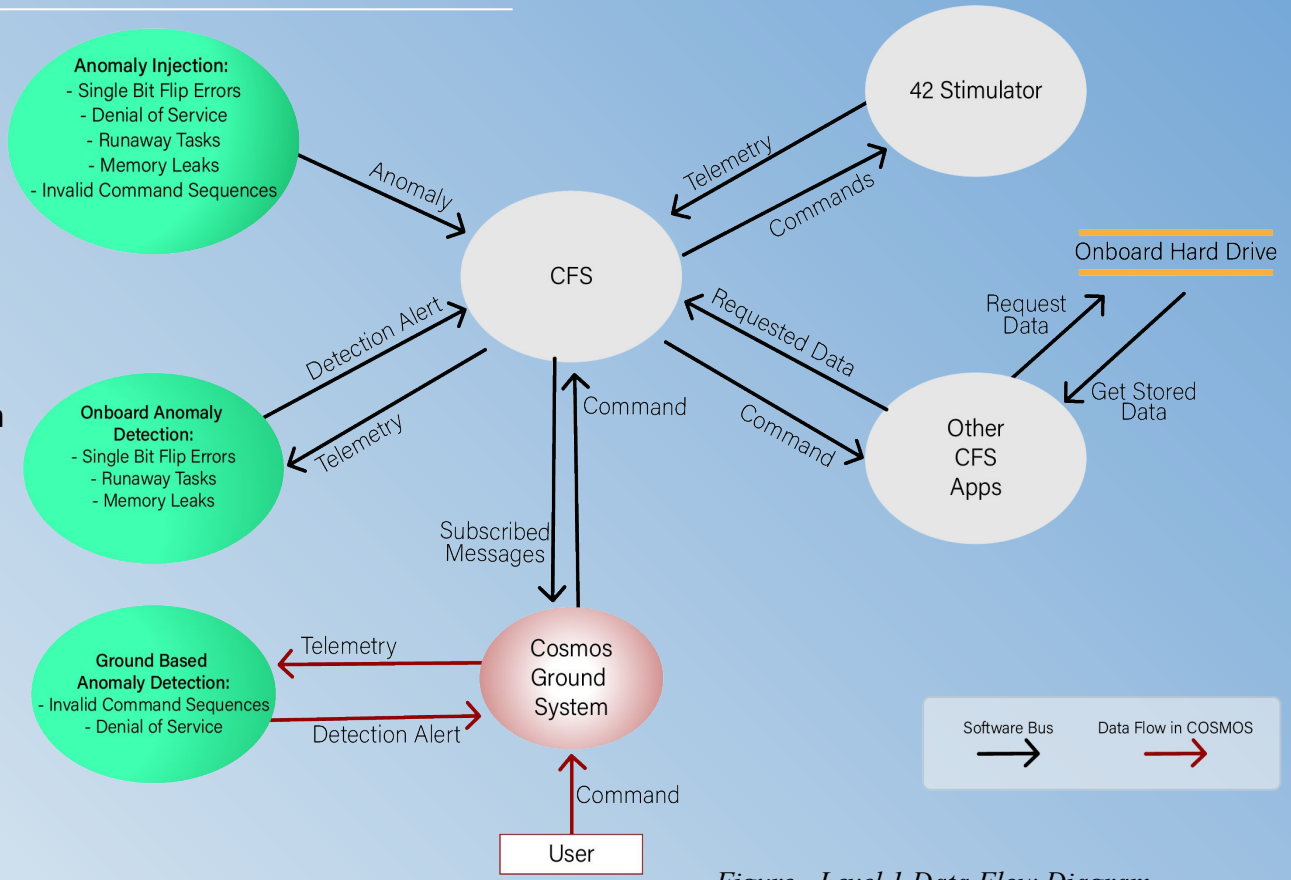
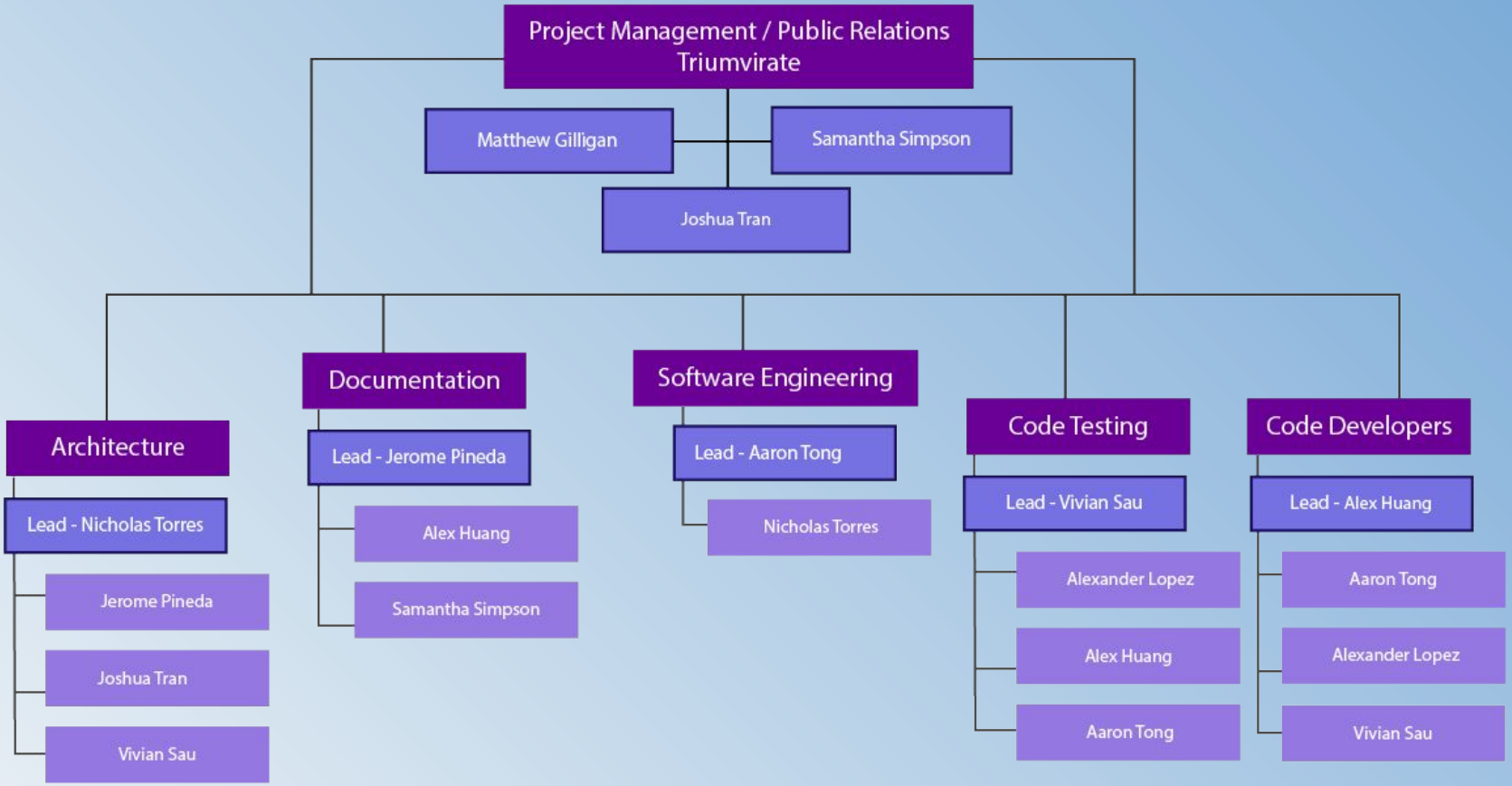


Figure - Level 1 Data Flow Diagram

Organizational Chart - Matthew



SRS document - Jerome

- Purpose
  - Supply information on the anomaly injection and detection requirements.
  - Provide information on simulation tools allows the users to configure and deploy platforms used in a real-time environment to simulate anomalies.

- Product Scope
  - The software products are Anomaly Injection, Onboard Anomaly Detection, and Ground Base Anomaly Detection.
  - All the systems will utilize the Open Sat Kit environment. The software will use the simulated telemetry data and satellite data to inject and detect anomalies.

SRS document - Jerome

- Main goals of this project:
  - Inject anomalous behavior/data
    - Use modern data analysis techniques/libraries/frameworks to detect those anomalies where traditional checks would fail

- Major technical hurdles of this project:
  - Detection of the anomaly either onboard the satellite and on the ground.
  - The anomaly detection made onboard the satellite processor has limited memory and CPU cycles running on a single thread.
  - The anomaly detection made for the ground has reduced response times due to data payload size when downlinking, as well as reduced opportunities to respond to the anomalies due to scheduling

Runaway task - Nicholas

- Injection
  - Create thread using pthead_create without calling join (make an orphan)
  - Thread will dynamically allocate memory
  - Thread executes computationally intensive tasks expected to slow system
  - The tasks are intentionally negative in order to see an impact in performance

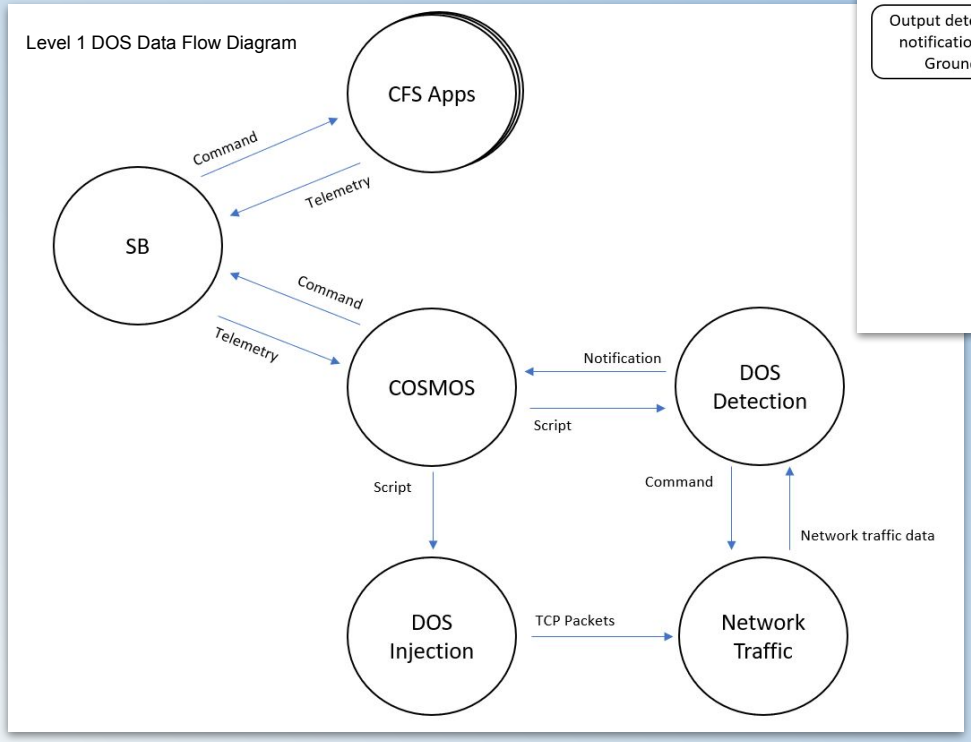
- Detection
  - All data storage tables are monitored (to see memory consumption)
  - Utilize Health and Safety applications to check resource consumption
  - Review all resource allocations
  - If resource exceeds overload consumption, ping anomaly
  - Else, keep running

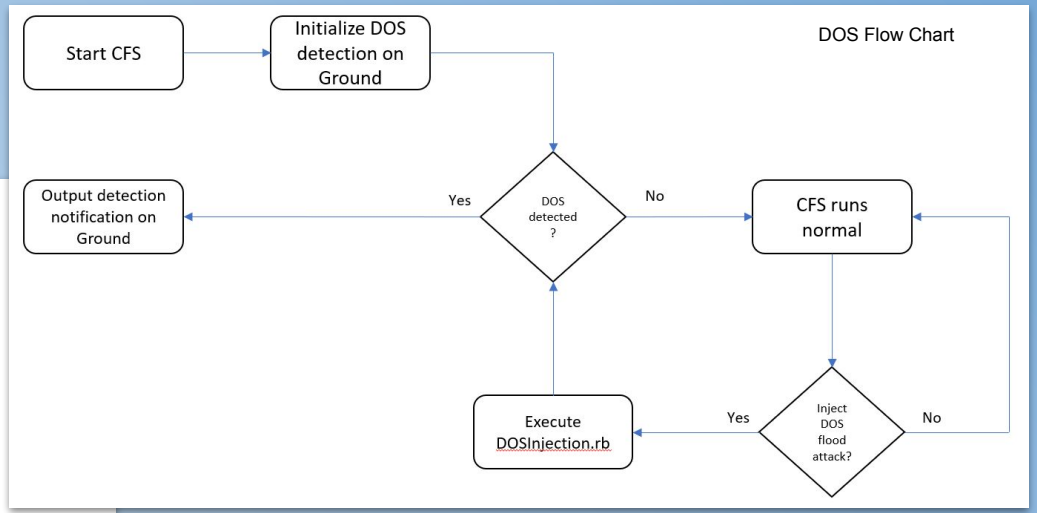
Memory leak - Alex Huang

- Injection
  - Create a dummy app that gets allocated a large section of memory
  - App gets assigned to CFs
  - Waits for user command to initiate dummy app process for anomaly injection
  - Certain memory allocation errors and performance issues expected to emerge after launch

- Detection
  - Memory Leak Detection
  - All data storage tables are monitored
  - Code runs to detect error count of specific errors, and also potentially track satellite status
  - If error count exceeds 10, ping anomaly
  - If satellite performance suffers, and discrepancies detected in memory allocation, ping anomaly
  - Alternatively, write a test function that checks to see if a certain amount of memory is available, if not then flag
  - Else, continue running

Denial of Service - Vivian





Invalid Command Sequence - Aaron

- Injection
  - Input commands that are impossible to execute
  - These commands will come from the Ground unit
  - The inability to be picked up as an error

- Detection
  - Run background command sequence detection by using error_counter from COSMOS table
  - If command is called on COSMOS
    - -Is command Successful?

    Yes: Run as commanded

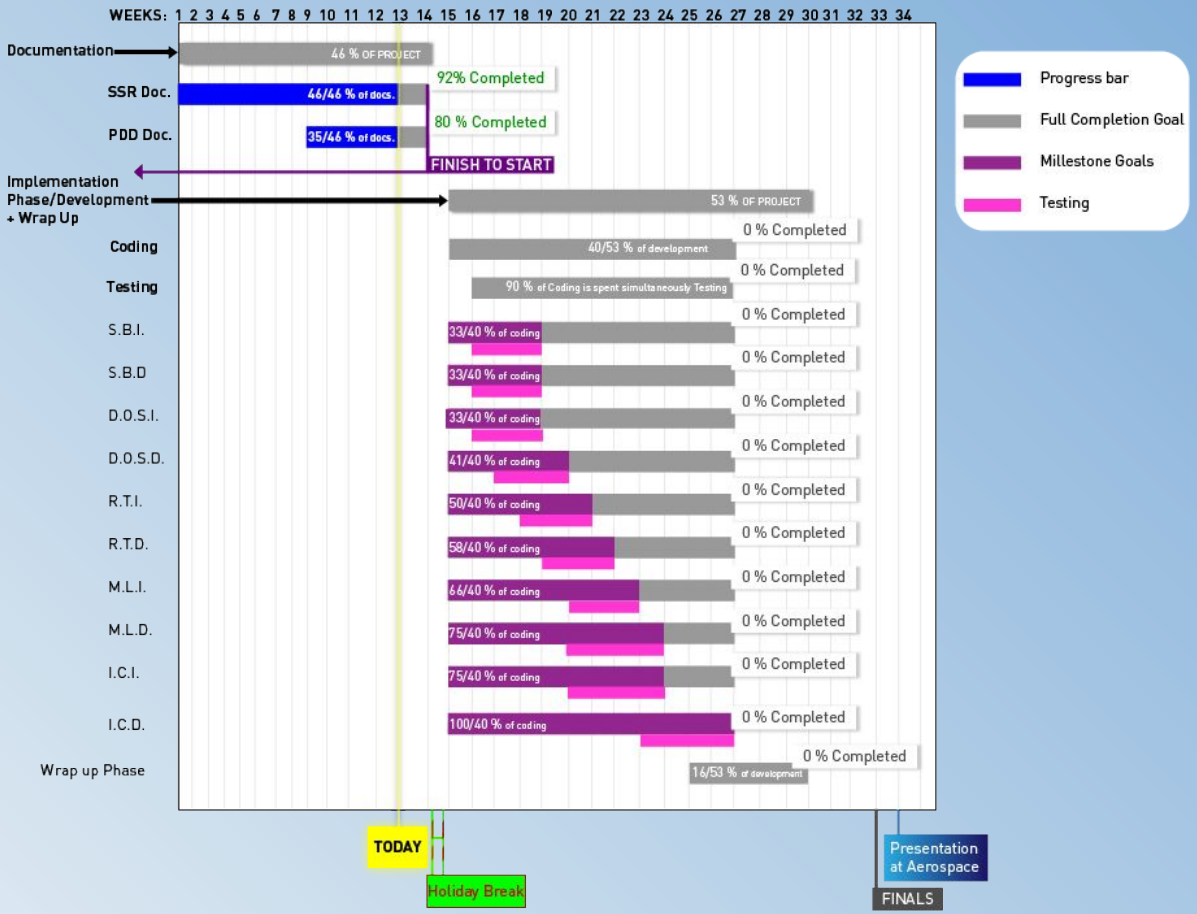
    No: Send Anomaly Notification

Single bit error - Alex Lopez

- Injection
  - MM Commands
    - Poke at any address
    - Create an area of disoriented Bits

- Detection
  - CCFDS File Delivery Protocol (CFDP)
  - Checksum
    - Considers expected data vs sent data
    - Directed toward telemetry tables

Project Management Review - Samantha



Fall																			
week starting on	19-Aug	24-Aug	31-Aug	7-Sep	14-Sep	21-Sep	28-Sep	5-Oct	12-Oct	19-Oct	26-Oct	2-Nov	9-Nov	16-Nov	23-Nov	30-Nov	7-Dec	14-Dec	21-Dec	28-Dec
week #			1	2	3	4	5	6	7	8	9	10	11	12	*Thanksgiving week*	13	14	*finals and break*		
meeting topic ***NEW***		Senior Design Launch at CSULA – Kick-off meeting			Status Update Telecon			**SRR/PDR & PMR a**			Status Update Telecon on **10/30**			**CDR & PMR**			Status Update Telecon on **12/11** End of doc. phase			
Milestone													-Injection side			-Detection side				
Winter																				
week starting on	4-Jan	18-Jan																		
week #	15	16																		
meeting topic ***NEW***	Start of Code Phase	Start of Testing Phase																		
Milestone																				
Spring																				
week starting on	25-Jan	1-Feb	8-Feb	15-Feb	22-Feb	1-Mar	8-Mar	15-Mar	22-Mar	29-Mar	5-Apr	12-Apr	19-Apr	26-Apr	3-May	10-May	17-May	24-May		
week #	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	*finals*			
meeting topic ***NEW***		Status Update Telecon on **02/05**			Status Update Telecon on **02/26**			Status Update Telecon on **03/19**	**TRR & PMR**		End of Code/Testing Phase Start of Wrap up Phase	Status Update Telecon on **04/30**		End of Wrap up Phase	Status Update Telecon on **05/07**			**Presentation at Aerospace**		
Milestone			-Single bit errors Injection/ Detection -Denial of service Injection	-Denial of service Detection	-Runaway Task Injection	-Runaway Task Detection	-Memory Leak Injection	-Memory Leak Detection -Invalid command sequences Injection			-Invalid command sequences Detection									

Milestones Updates - updated on 12/11

- DOCUMENTATION
  - Software Requirements Specification - completed
  - Critical Design Document - 12/7 12/20

- DEVELOPMENT
  - Ground-Zero Prototype - 12/11 - completed
    - Basic implementation of Denial of Service Injection and Detection systems
    - The development team currently working on the injection system

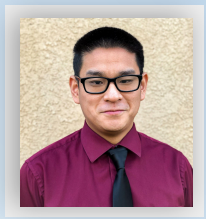
  - Single bit errors injection - 2/12 2/22 detection - 2/12 2/22
  - Denial of service injection - 2/12 12/24 detection - 2/19 12/24
  - Runaway tasks injection - 2/26 2/26 detection - 3/5 3/5
  - Memory Leaks injection - 3/12 3/12 detection - 3/19 3/19
  - Invalid command sequences injection - 3/19 3/12 detection - 4/9 4/9

Ethics - Joshua

- Issues
  - The anomaly injection tool, as stated by Aerospace, will be breaking regulatory protocols that will lead into some kind of damage. This damage goes against the ACM Code of Ethics section 1.2: Avoid Harm. In this section avoiding harm is “negative consequences especially when those consequences are significant and unjust.” With the anomaly injection tool we are creating the potential to do unjustified damage to property
  - These anomalies all break the protocol and cripples the effectiveness of the satellite

- Safeguards
  - Anomalies are only injected by the user's knowledge
  - Automated system is also controlled by the user in terms of an on and off switch
  - Failsafe method that will revert the satellite to its original functioning state

Thank you,
The Team

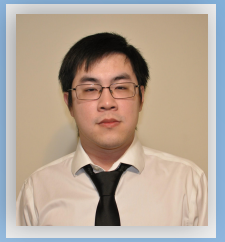


Joshua Tran
josh.dtran@gmail.com

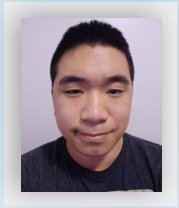
Samantha Simpson
simpsonnsamantha@yahoo.com

Matthew Gilligan
matthew.gilligan98@gmail.com

Nicholas Torres
torresnick272@gmail.com

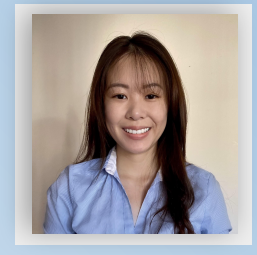


Alex Huang
interdimensionalwaterbear@gmail.com

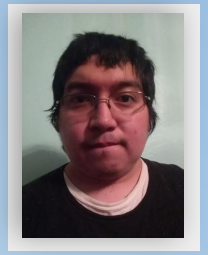


Aaron Tong
pbjt2.edu@gmail.com

Jerome Pineda
jeromepineda79@gmail.com



Vivian Sau
sauvivian@gmail.com



Alex Lopez
alalexalex152@gmail.com