



Applying Modern Software System Design to Small Satellite Development and Operations

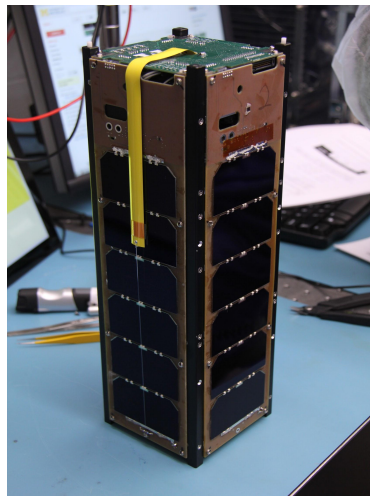
Michael Wilson

Michigan eXploration Laboratory - University of Michigan

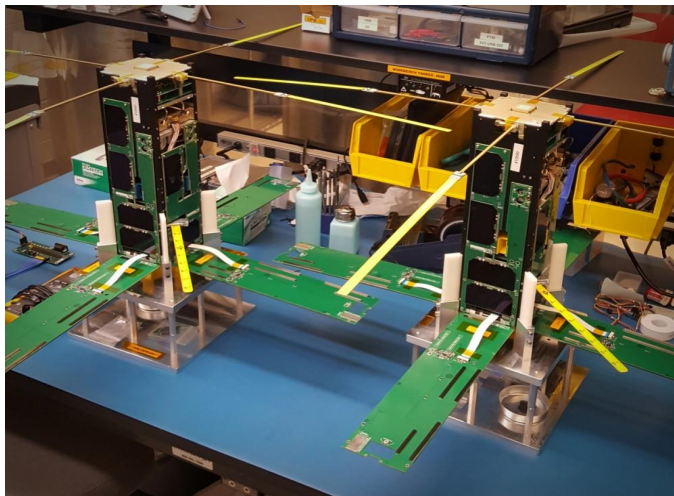
34th Annual Small Satellite Conference

Michigan eXploration Laboratory

Fall 2018: Our previous operations system was lost due to backup failures, but we were in the midst of developing and operating 3 flight spacecraft and 3 engineering development units.



GRIFEX

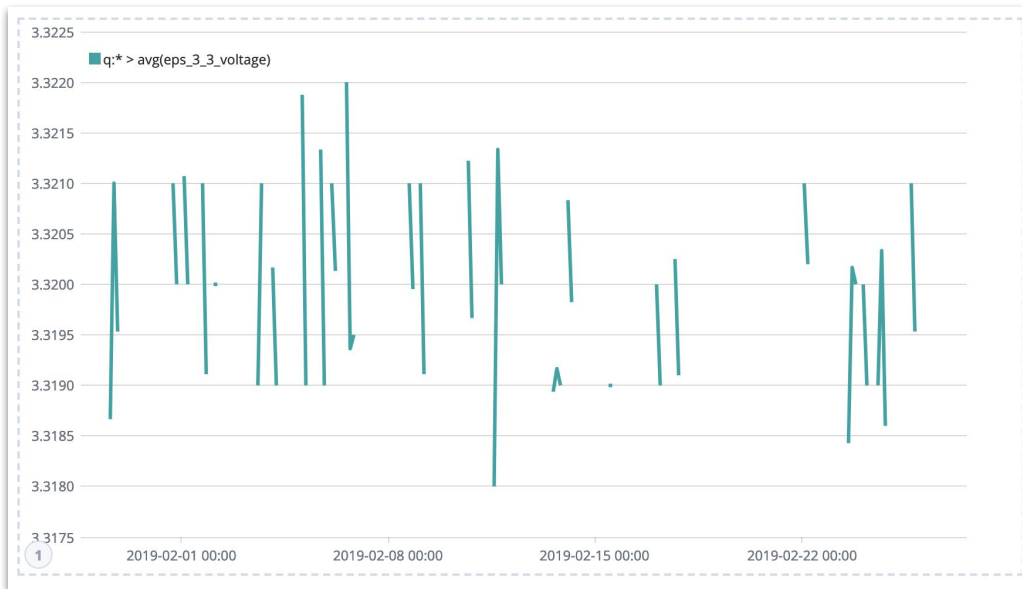
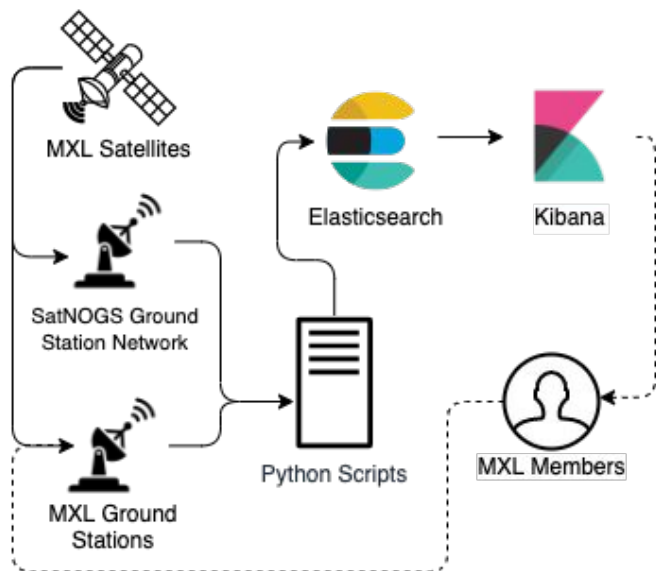


E-TBEx



MC 9

Initial work - Python scripts create simple graphs



Python scripts pulled data from the SatNOGS API and our ground station logs and dumped packets into Elasticsearch, where they could be viewed using a Kibana dashboard.



System Goals

An effective and lasting solution should enable the following:

- Reliable, thorough testing
- Rapid, stable, feature development
- Knowledge transfer between students
- Task automation
- Operational visibility

These goals are achieved using *modern software system design*.



Modern Software System Design

3 Core Principles

- Small - applications are broken down into granular, easily tested, components
- Networked - applications can communicate in a distributed manner and are therefore more resilient in the face of failures and easier to manage at the service level
- Developer Oriented - applications follow simple architectures to which engineers can easily contribute and deliver changes

The application of these principles toward a distributed satellite operations and development system resulted in the creation of the MXL Integrated Data Analysis System (MIDAS).





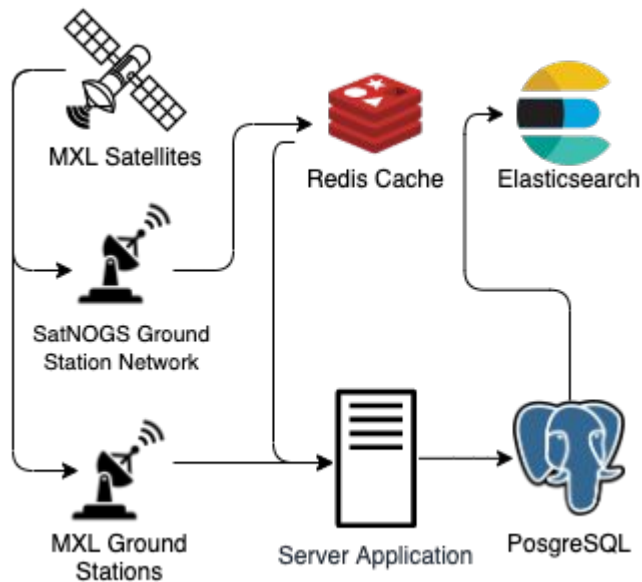
Areas of Improvement

- Telemetry aggregation and analysis
- Ground systems and satellite communications
- Systems monitoring and development
- Team communications and collaboration

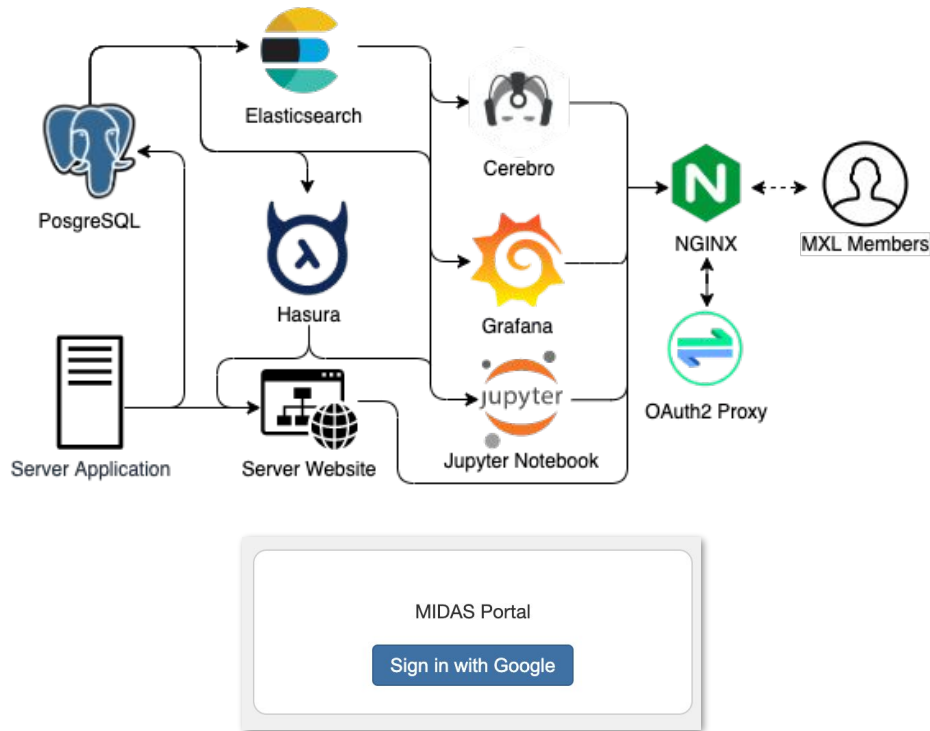
Telemetry aggregation and analysis - Architecture

The server aggregates telemetry data from the SatNOGS API and our own ground stations.

MIDAS *extracts* data from these sources, *transforms* it, and *loads* it into the PostgreSQL and Elasticsearch data stores (ETL Pipeline).



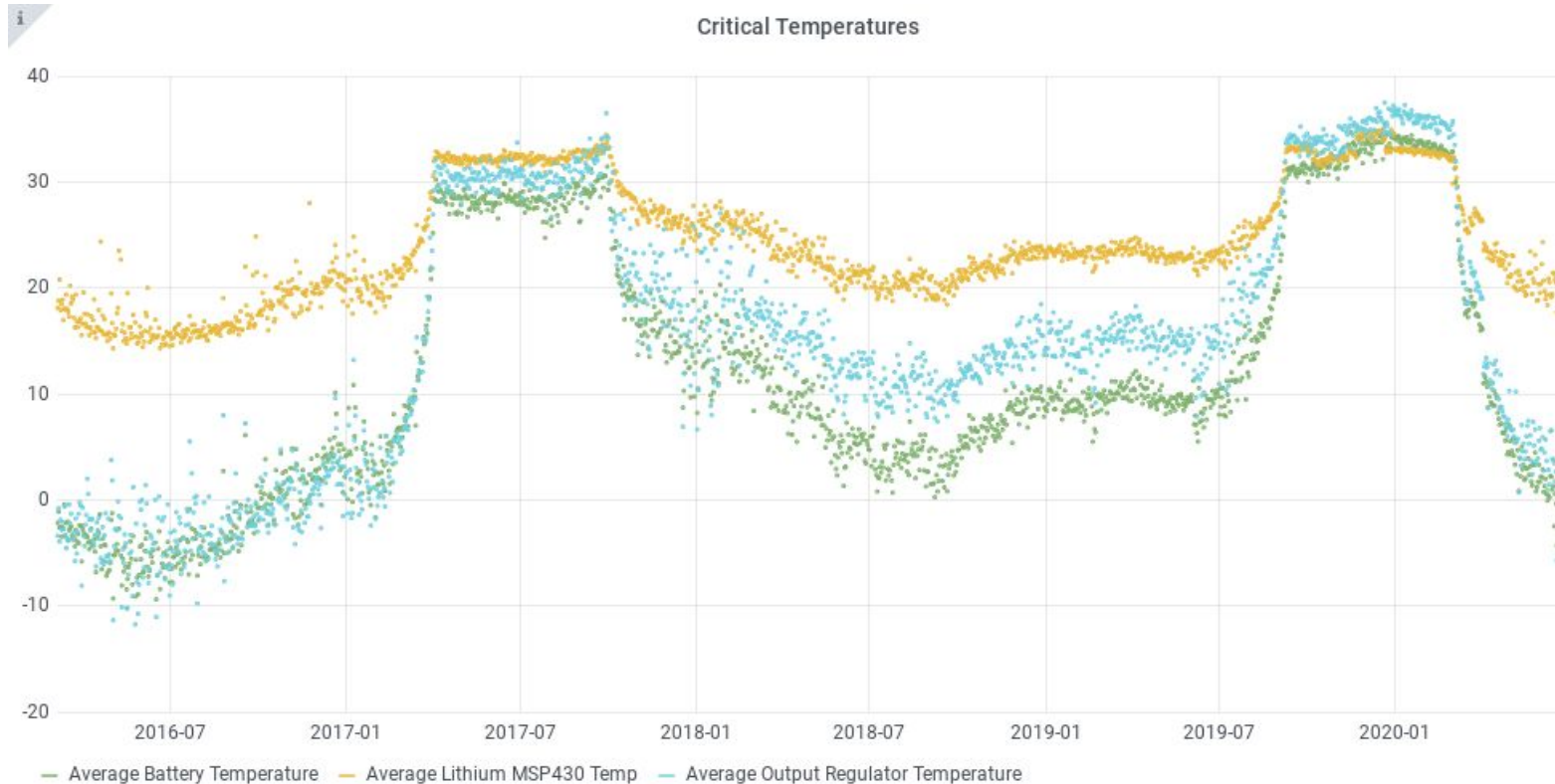
Telemetry aggregation and analysis - Architecture



MIDAS services including Grafana, Hasura, Jupyter Notebook, Cerebro, and our own custom dashboards query and analyze data from PostgreSQL and Elasticsearch.

User requests are authenticated using University of Michigan Google accounts and proxied to their appropriate services, e.g. <https://grafana.mxl-ops.engin.umich.edu/>

Telemetry aggregation and analysis - Grafana





Telemetry aggregation and analysis - GraphQL

MIDAS's Hasura service provides users with a GraphQL endpoint that enables intuitive queries on relational and graph data.

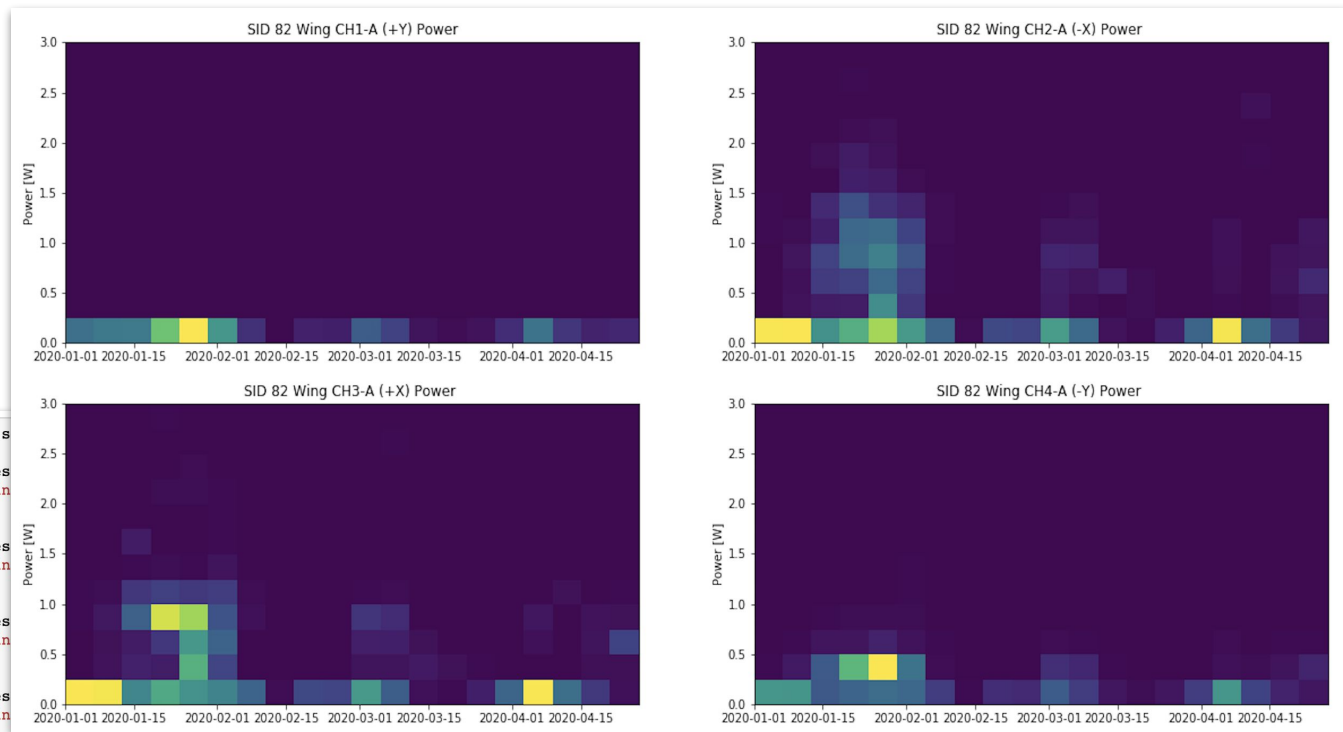
```
query MyQuery {  
  mxl_pass(  
    where: {  
      satnogs_id: {_is_null: false},  
      packets: {packet: {flag: {_eq: "1"}}}  
    },  
    order_by: {aos: desc},  
    limit: 5) {  
    aos  
    ground_station { name }  
    craft { name }  
    packets_aggregate {  
      aggregate {  
        count  
      }  
    }  
  }  
}
```

```
[{  
  "aos": "2020-06-16T17:05:52+00:00",  
  "ground_station": {  
    "name": "CA,USA-Fixed UHF"  
  },  
  "craft": {  
    "name": "GRIFEX"  
  },  
  "packets_aggregate": {  
    "aggregate": {  
      "count": 99  
    }  
  }  
}]
```

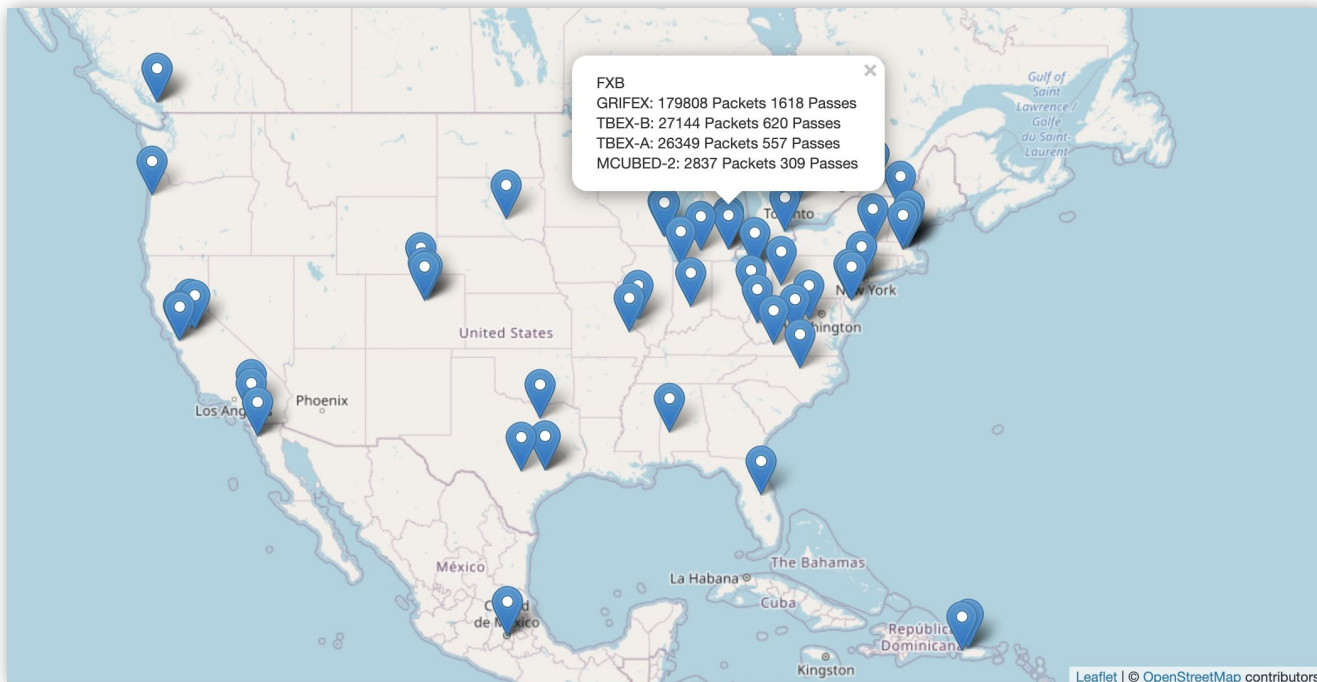
Telemetry aggregation and analysis - Jupyter

To perform extended analysis, users load data into Jupyter Notebook, where they can manipulate it using Python.

```
In [6]: 1 fig, [[ax1,ax2],[ax3,ax4]] = plt.s
2
3 ax1.pcolor(xedges1_datetime,yedges
4 ax1.set_title('SID ' + sid + ' Win
5 ax1.set_ylabel('Power [W]')
6
7 ax2.pcolor(xedges2_datetime,yedges
8 ax2.set_title('SID ' + sid + ' Win
9 ax2.set_ylabel('Power [W]')
10
11 ax3.pcolor(xedges3_datetime,yedges
12 ax3.set_title('SID ' + sid + ' Win
13 ax3.set_ylabel('Power [W]')
14
15 ax4.pcolor(xedges4_datetime,yedges
16 ax4.set_title('SID ' + sid + ' Win
17 ax4.set_ylabel('Power [W]')
18
19 plt.savefig(sid+ ch + ' Wing Panel Power 2D Histogram_' + startString + '_' + endString + '.png')
20
```

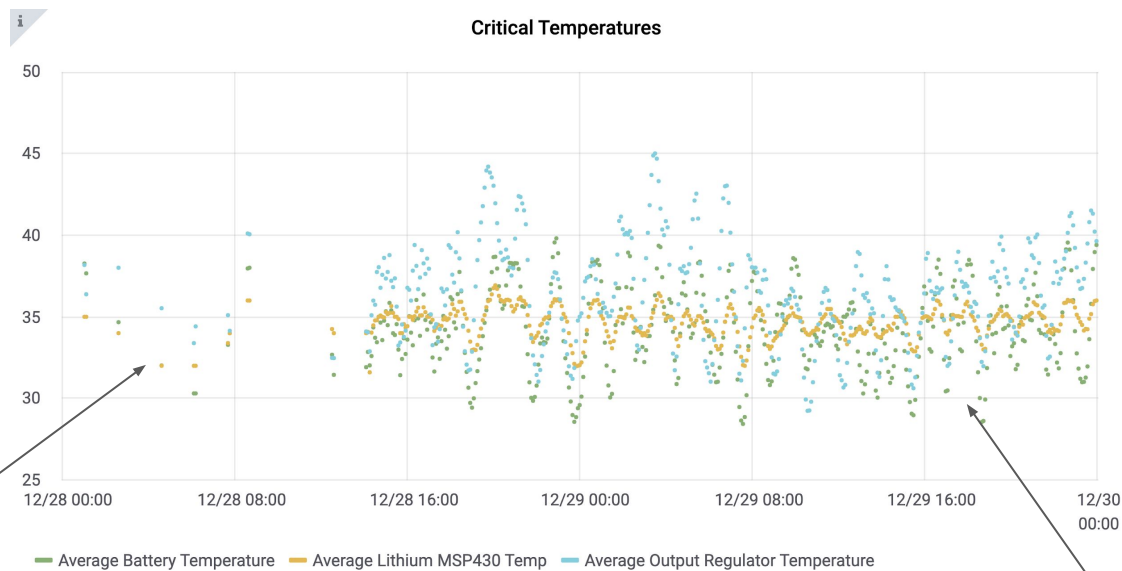


Ground systems and satellite communications



MIDAS integration between MXL ground stations and SatNOGS gives us uniform data across 100s of locations around the globe.

Ground systems and satellite communications






Several Dozen
Beacons Every Day

Downlinked telemetry archives provide
a **125x** increase in data resolution.

10 Second Resolution
Beacon Telemetry

Ground systems and satellite communications

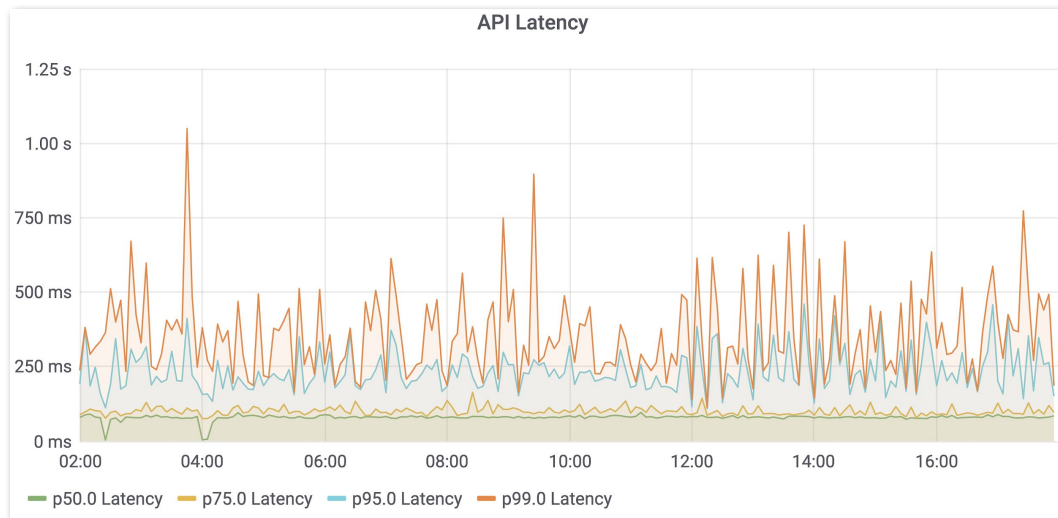
TBEX-A						
File Number	Total File Parts	MD5 Sum	Progress	Download	Telemetry	Archive
125	1	CA0E115E03F1D3FE6A3963E8BB2C603D	<div><div></div></div> 1/1		<input type="radio"/>	
104	29	7D6C98EBBEB8977BBC1B107DAF3FB7D5	<div><div></div></div> 28/29	-	<input type="radio"/>	
103	64	758FC420552BFC4C5CFC92EC8ACDCDA3	<div><div></div></div> 35/64	-	<input type="radio"/>	
102	9	4EA5969BFA96A3D6405BA7344CFF2C7E	<div><div></div></div> 9/9		<input type="radio"/>	
101	2	3F6484A6EEA96AD4C575F2E15CF0A92E	<div><div></div></div> 2/2		<input type="radio"/>	
39	21	3218003C4097EBD776072D4E6382079D	<div><div></div></div> 21/21		<input type="radio"/>	

Custom data displays show file download progress and an MD5 sum for checking data integrity.

Systems monitoring and development - Grafana

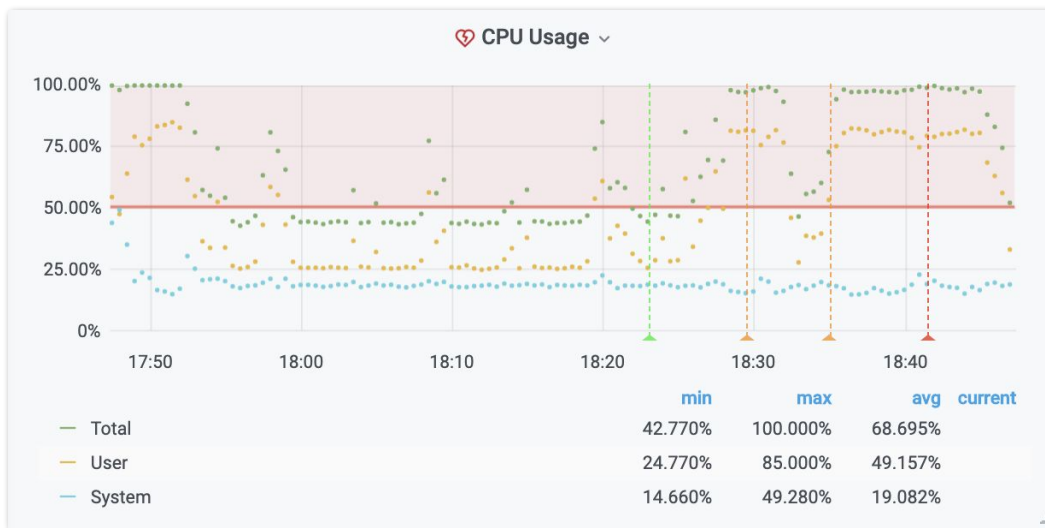
Logs and metrics from all MIDAS Docker containers are pushed into Elasticsearch, enabling high operational visibility.


Just as we have telemetry and operations for our spacecraft, we now have them for our ground systems as well.



Systems monitoring and development - Grafana

Monitoring metrics with such tools as Grafana allows the creation of alert systems that notify our team via Slack in the event of system errors, whether on the ground or in orbit.





MXL Grafana APP 2:48 PM

[Alerting] CPU Usage Alert








































[Alerting] CPU Usage Alert

Total

0.79997000396252

 Grafana v7.0.1 | Today at 2:48 PM

Systems monitoring and development - GitLab

	#15269		 <code>python-clie...</code>  <code>622f25f0</code> 	 00:01:02  6 days ago	
	#15265		 <code>master</code>  <code>4a46946c</code> 	 00:04:12  6 days ago	
	#15263 		 <code>13342-tbex-...</code>  <code>2df6e41b</code> 	 00:01:25  1 week ago	
	#15262		 <code>13342-tbex-...</code>  <code>c8be44b8</code> 	 00:24:00  1 week ago	
	#15261 		 <code>14223-add-3...</code>  <code>5f1f35b1</code> 	 00:01:14  1 week ago	

GitLab pipelines allow us to perform automated end-to-end tests, periodically run data backups, and ensure that any software changes are validated before being put into production on the ground, or in orbit.



Team communications and collaboration - Slack



MIDAS APP 8:27 PM

RAP 2f01af87-d52c-5dcc-9a24-2a187d65fc13 Type CAP PID 0 SID 66 Received At 2020-05-03T00:27:50Z HMAC CAC52D45:

CAP ID 9 Run Linux Command Subsystem ID 1 Ref Number: 0

Arguments:

1 Linux System Call (String): sh /dev/datamnt/exec/dl_script_c.sh

/dev/datamnt/downlink/Beacon_Archive_992.tar.gz 26647 1 1000 211 420



GitLab APP 12:28 PM



Michael Wilson (mewil)

Pipeline #15078 has failed in 09:18

Branch

master

Failed stage

backup

Commit

Variable for jwc custom prints.

Failed job

midas_server_backup

MXL Software | May 19th



GRIFEX Passes APP 9:57 PM

Event starting in 10 minutes:

GRIFEX Pass

Today from 10:07 PM to 10:18 PM

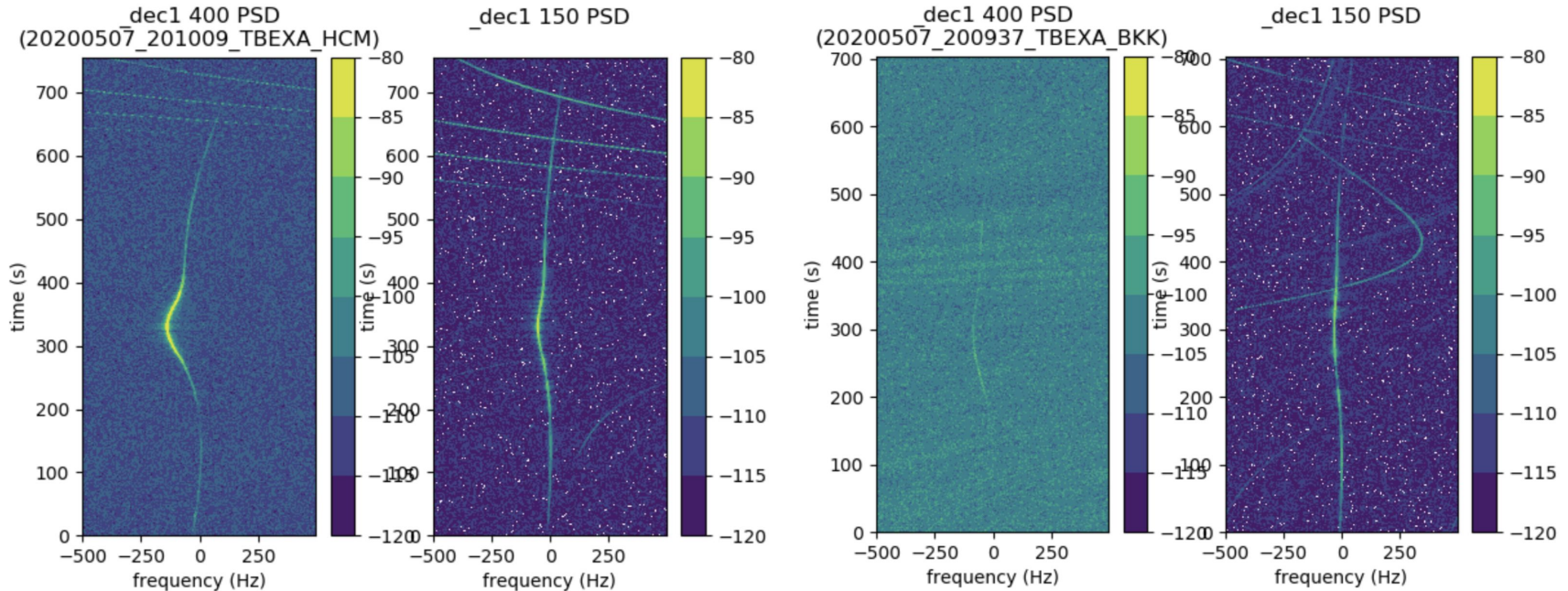




Conclusions

- By using modern software system design—keeping our applications small, networked, and developer oriented, MIDAS has undoubtedly transformed MXL satellite operations for the better.
- Automation, monitoring, and communication tools have enabled remote work during the COVID-19 pandemic.
- Our work clearly illustrates that modern software solutions that emphasize system autonomy and measurability have the power to dramatically improve small satellite operations and development.
- Next steps include automated command campaigns to ease the load on our operators and migrating our servers to the cloud to decrease maintenance time.

MIDAS Scheduled E-TBEx Operations In Bangkok and Ho Chi Minh



Questions?