



---

---

# TAT-C: A Trade-space Analysis Tool for Constellations

Jacqueline Le Moigne,  
Philip Dabney, Olivier de Weck,  
Veronica Foreman, Paul Grogan,  
Matthew Holland, Steven Hughes, Sreeja Nag

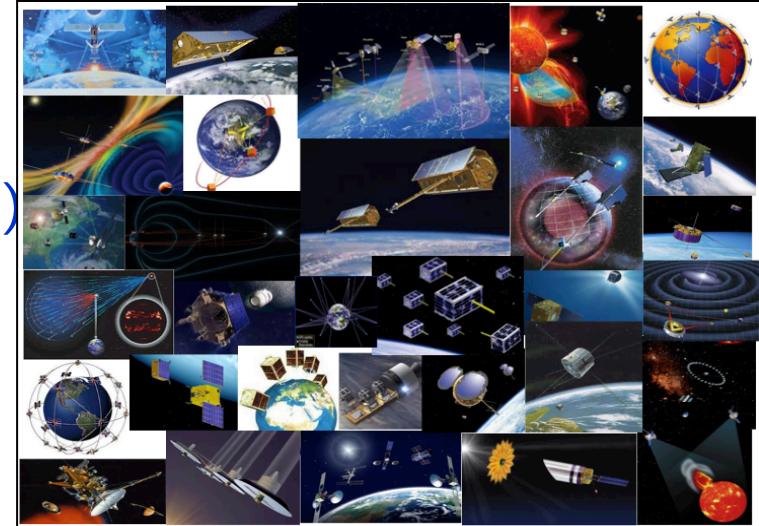
AIST-14-0053 – ESTF16  
June 14, 2016



# DSM Background

## What is a Distributed Spacecraft Mission (DSM)?

A Distributed Spacecraft Mission (DSM) is a mission that involves multiple spacecraft to achieve one or more common goals.



## Driver/Need

- Reduce the cost, risk and implementation schedule of all future NASA missions
- Investigate the minimum requirements and capabilities to cost effectively manage future multiple platform missions and to cost effectively develop and deploy such missions

## Previous Study

- Determine the Science relevance and the main technology challenges associated with DSM
- Collected data on 62 missions (Earth Science: *12 missions – 7 operational, 3 in development and 2 concepts*)



# Some Earth Science DSM Applications

---

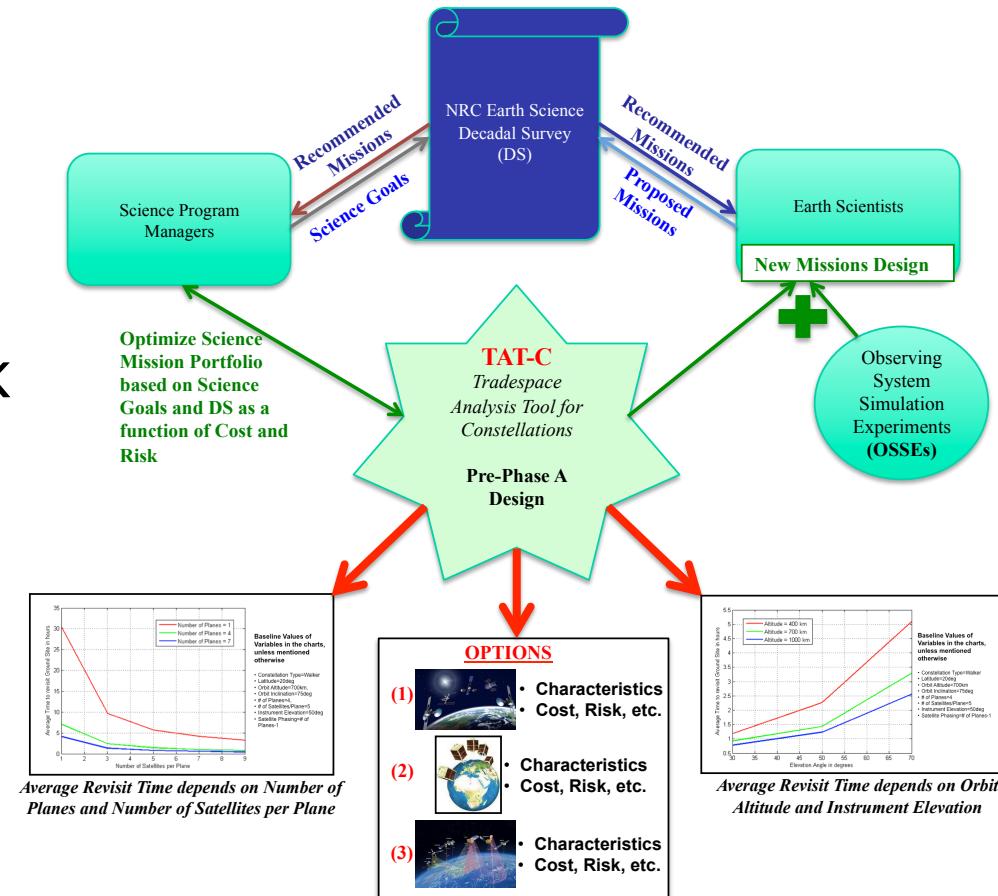
---

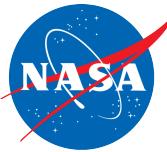
- Studying the Earth Radiation Budget (e.g., CLARREO proposed as a constellation of 6 satellites) with:
  - Better radiometric accuracy
  - Multi-angle view data
- Increasing the temporal and spatial resolution (e.g., Landsat-like constellation)
  - Time series
  - Physiology of plants and phenology of vegetation
- Combining information from complementary sensors (similar to A-Train)
- Investigating various orbits



# Benefits of DSM

- Enabling New Measurements
- Improving Existing Measurements
- Mitigating Missions Risk
- Reducing Costs
- Facilitating Data Continuity
- Enabling Multi-Organization and International Cooperation





# Objectives and Overall Description

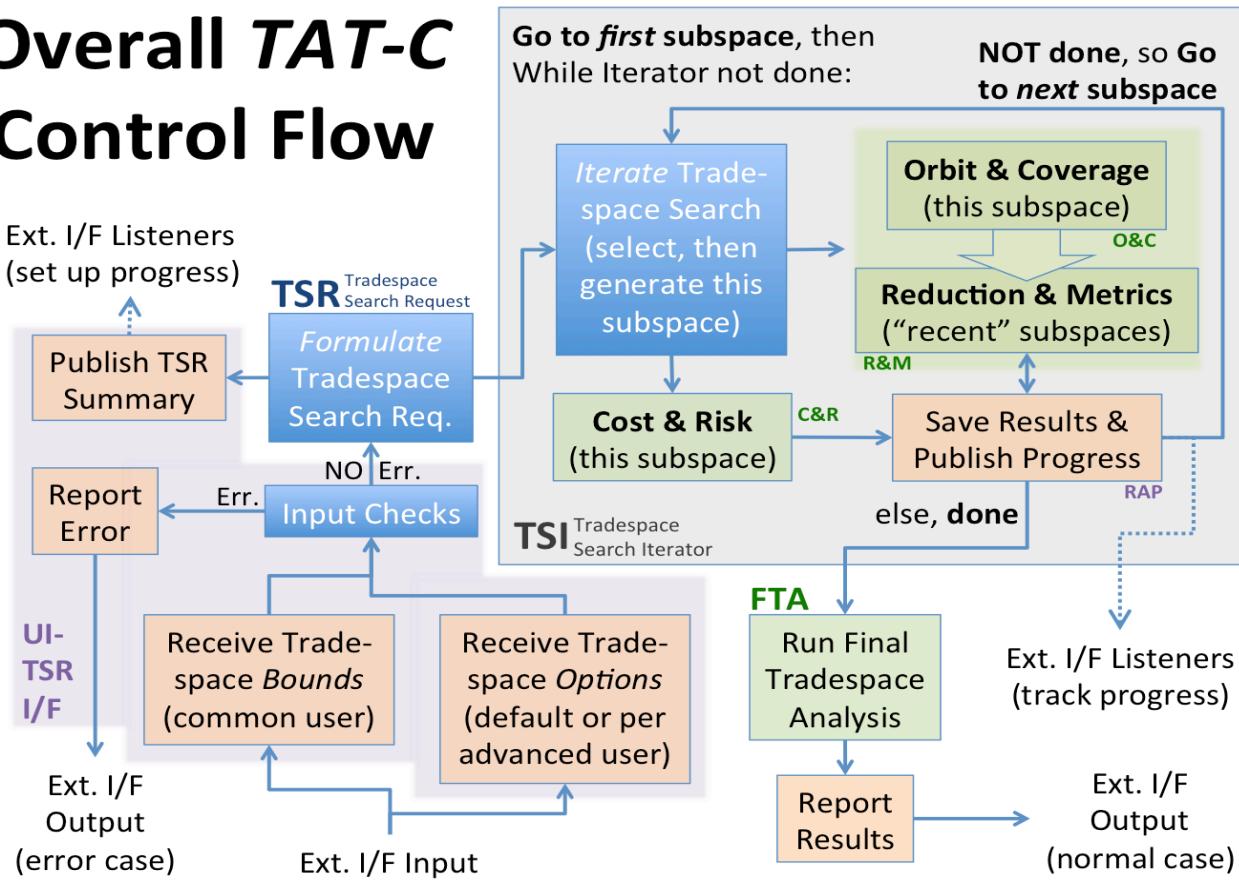
---

---

- Provide a framework to perform pre-Phase A mission analysis of Distributed Spacecraft Missions (DSM)
  - Handle multiple spacecraft sharing mission objective
  - Include sets of smallsats up through flagships
  - Explore trade space of variables for pre-defined science, cost and risk goals, and metrics
  - Optimize cost and performance across multiple instruments and platforms vs. one at a time
- Create an open access toolset which handles specific science objectives and architectures
  - Increase the variability of orbit characteristics, constellation configurations, and architecture types
  - Remove STK or any commercial licensing restrictions

# TAT-C Architecture

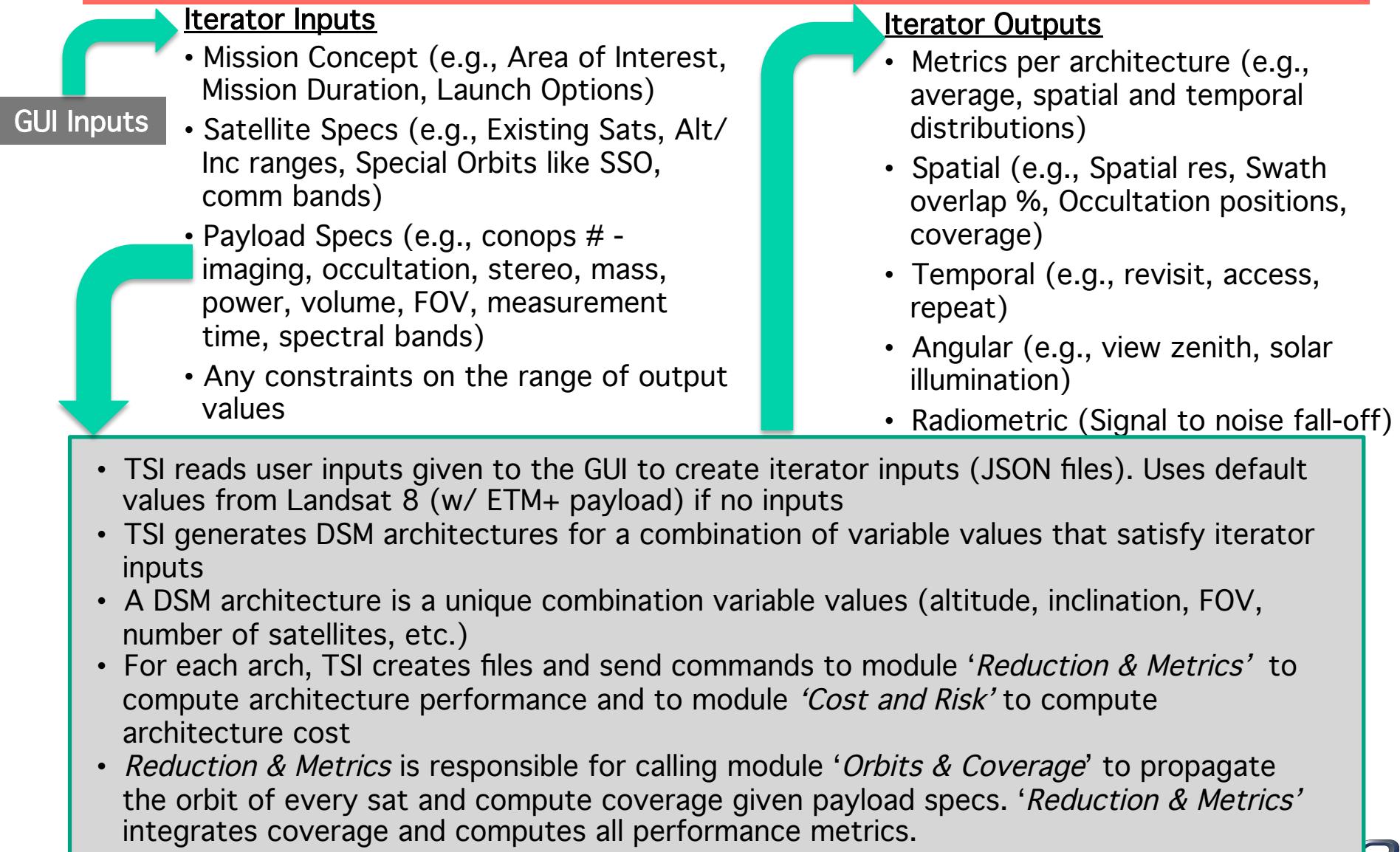
## Overall TAT-C Control Flow



- Currently, TAT-C can generate architectures for 3 types of missions/payloads: imagers, occultors (to occult with fixed bodies like the Sun or moving bodies like the GPS) and pair-wise functions (bi-static radar, stereo)
- Default ground stations are DSN, NEN w/ positions, bands.
- Launch vehicles database available with cost, reliability, frequency



# Tradespace Search Iterator (TSI)





# Tradespace Iterator

---

## Constellation Types (Currently):

- Uniform Walker Constellations – all sats in the same alt, inc and equal sats per plane
- Ad-Hoc Constellations – all sats assumed to be located at one of the 48 current Planet Labs slots (TLEs available online)

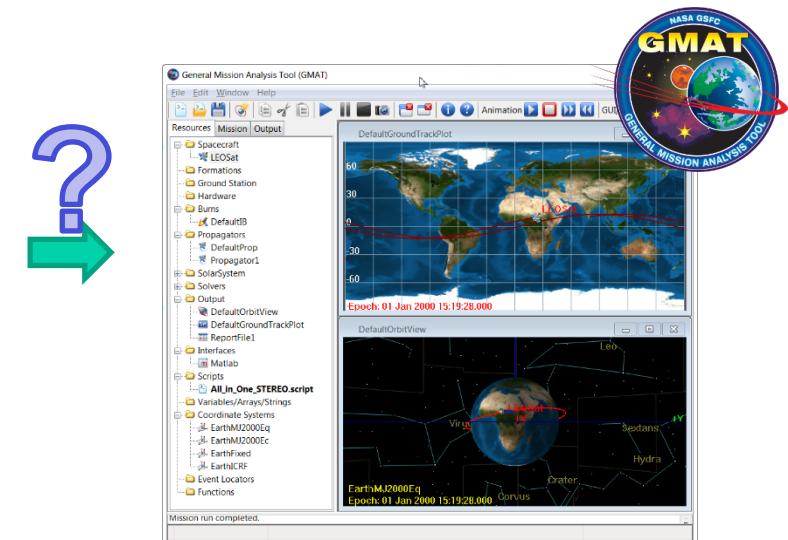
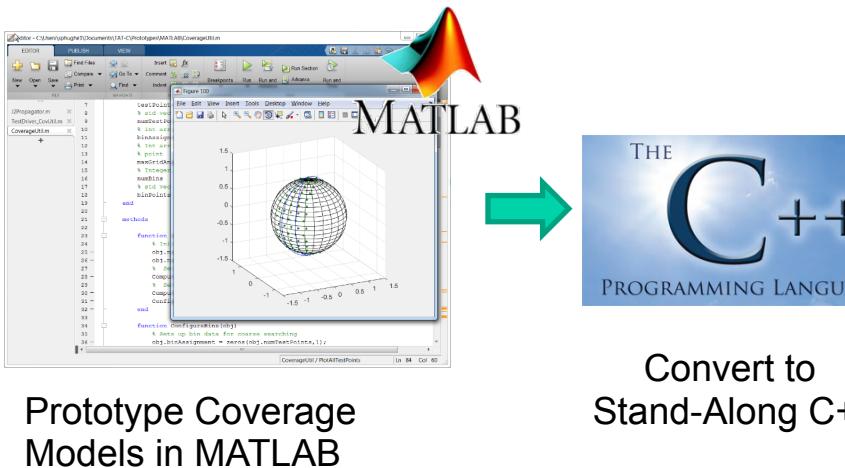
## Constellation Types (In Development):

- Non-Uniform Walker constellations – combinations of uniform Walker constellations over diff alt, inc, sats/plane
- Ad-Hoc Constellations where sats are launched as per the next available launch option from launch D/B
- Precessing constellation where sats are dropped off by a single LV at differential alt, inc and disperse over time to give RAAN coverage



# Orbit-Coverage Module

- Purpose of Module
    - Model orbits balancing accuracy and performance
    - Compute coverage metrics for constellation/sensor set
    - Compute ancillary orbit data for performance, cost, risk
  - Development Approach



Potentially integrate into GMAT  
depending upon future needs



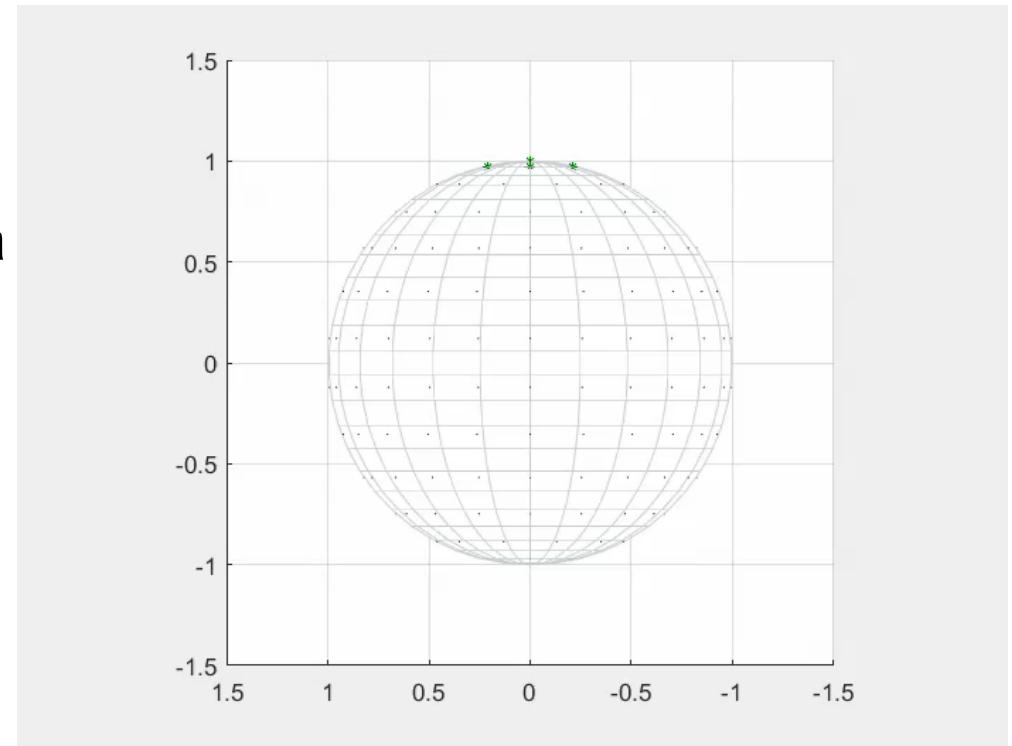
# Orbit-Coverage Module (cont.)

- Concept:

- TSI Manages mission architecture configuration
- TSI Calls O&C to compute coverage and ancillary data
- O&C returns that information to TSI/R&M for further analysis and reduction

- Capabilities:

- Evenly spaced grid points
- Custom grid points
- Conical sensor
- J2 Dynamics (fast)
- Multiple spacecraft



Graphic Generated by Coverage Prototype



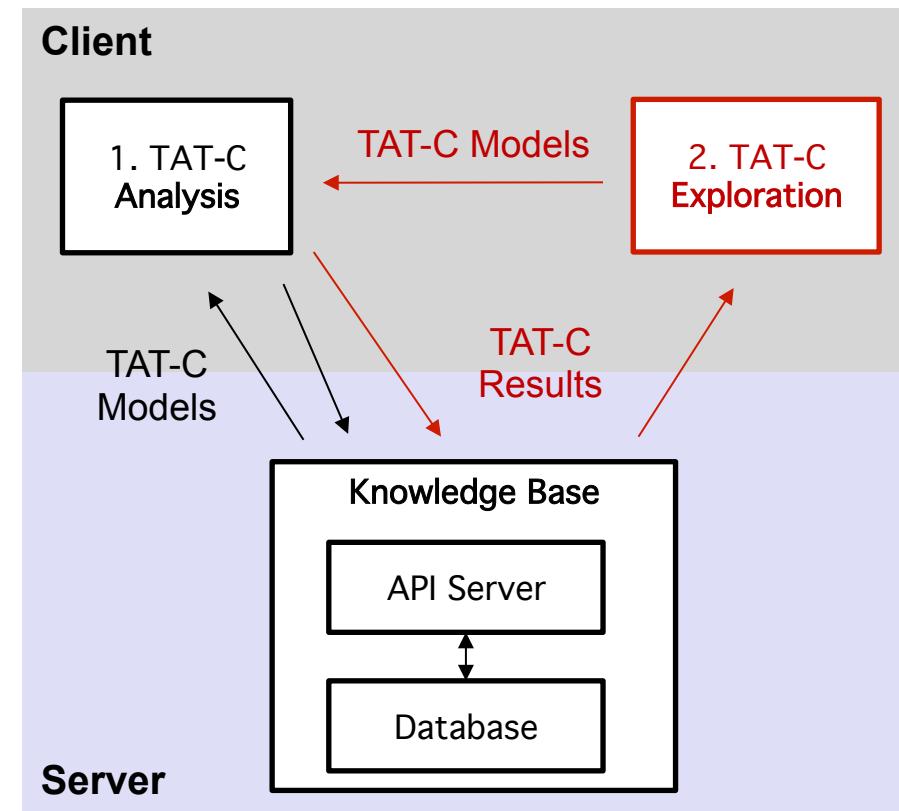
# Value, Cost and Risk Module

---

- Given a satellite constellation architecture, the VCR module will provide estimates of:
  - Value, expressed in dollars or utility
  - Cost, life cycle cost (RDT&E, manufacturing, launch, operations)
  - Risk, profile of the system technical and cost risk
- Traditionally, cost and risk estimates have been focused on individual satellites and fixed architectures.
  - It can be difficult to integrate these estimates with the decision making process
  - Previous work has sought to identify the limits of the current tools as they pertain to DSMs; we have continued and extended this effort
- VCR Module will enable trades between performance and value/cost/risk more readily
  - Current work: developing a transparent cost estimating approach that aggregates existing tools and adapts them to DSMs
  - Future work: development of risk and value elements

# Knowledge Base

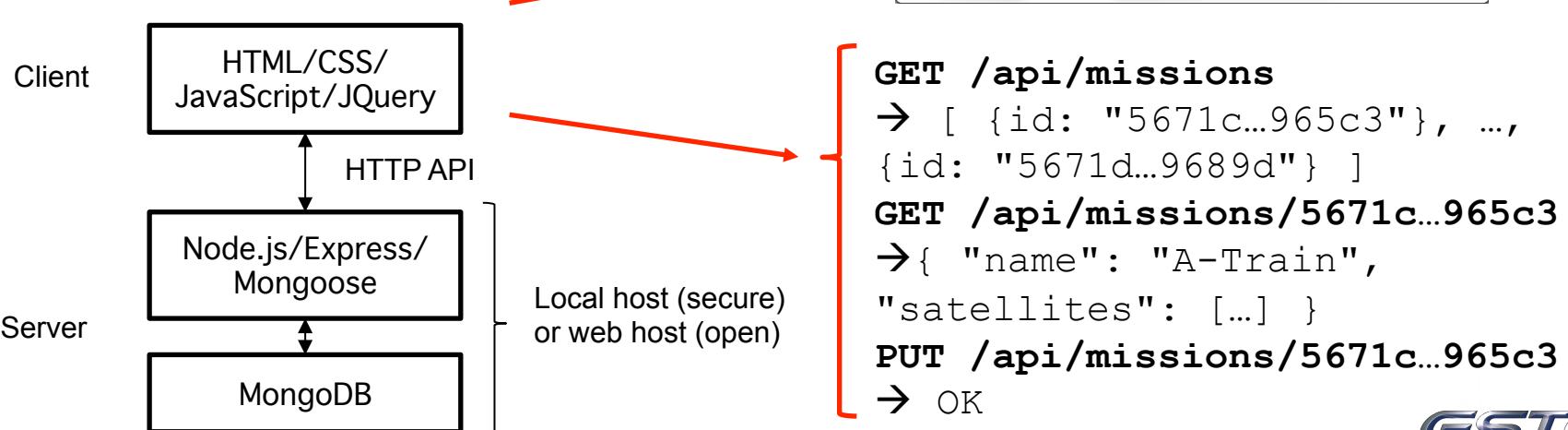
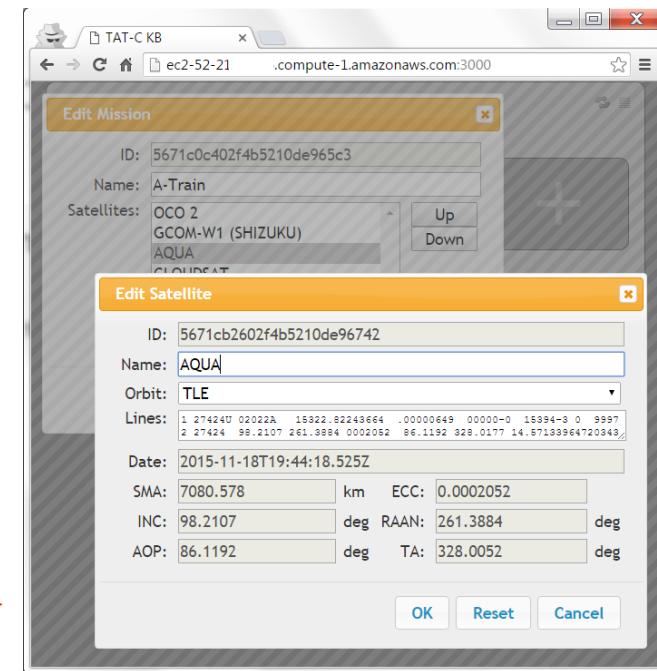
- Centralized store of structured data
- Support TAT-C tasks:
  1. Analysis: compose new mission concepts from existing models
  2. Exploration: discover new mission concepts by querying previous results





# Knowledge Base Prototype

- Demonstrate technical feasibility
  - HTTP requests (JSON)
  - Browser-based GUI client
- Layered client-server architecture
- GET or PUT models:
  - Mission > Satellite > Orbit





# User Interface

---

Currently Planned TAT-C User Interface includes GUIs, CLIs and APIs.

**GUIs (Graphical User Interfaces)** will be portable to any typical graphical computing environment, and will be designed to function like familiar “Software Wizards”, walking users systematically through DSM tradespace choices, and their consequences.

- GUIs will be designed to isolate basic, required, non-expert choices from more expert options typically accessed by more advanced users.
- GUIs will intuitively blend interactive choosing with visual browsing of analysis output characterizing the results of choices.

**CLIs (Command Line Interfaces)** will be portable to typical command-line environments, and will be designed to enable scripting of interactions equivalent to those possible via GUIs, especially once users establish (and want to automate) their preferred workflows.

**APIs (Application Programmer Interfaces)** will expose internal software interfaces to skilled programmers with the expert ability to develop software applications in the “TAT-C Ecosystem”.



# Graphical User Interface

● ● ● **Bounds of the TAT-C Trade Space**

Max. cost [\$]:	<input type="text" value="0"/>
Max. number of observatories [#]:	<input type="text" value="0"/>
Min. total time spent observing [s]:	<input type="text" value="0"/>
Max. total time spent observing [s]:	<input type="text" value="0"/>
Max. coverage time [s]:	<input type="text" value="0"/>
Max. revisit time [s]:	<input type="text" value="0"/>
Min. integrate time [s]:	<input type="text" value="0"/>
Max. downlink latency [s]:	<input type="text" value="0"/>
Max. spatial resolution [m]:	<input type="text" value="0"/>
Min. altitude [m]:	<input type="text" value="0"/>
Max. altitude [m]:	<input type="text" value="0"/>

[Set Advanced Options](#) [Run Tradespace Analysis](#)



# Graphical User Interface

The screenshot shows a graphical user interface for setting advanced options for a tradespace search request. On the left, there is a panel titled "Bounds of the tradespace" containing various input fields for constraints like cost, number of observatories, and time spent observing. In the center, a tree view displays the hierarchical structure of the tradespace search request, with nodes for top-level windows, panels, notebooks, tree controls, search controls, text controls, status bars, frames, and inspection frames. On the right, a dynamic view shows examples of how specific widgets are being configured, including set value, set digits, and set format options for floating-point numbers. At the bottom, there are buttons for "Set by Simple Bounds", "Load TSR", "Save TSR", and "Run Tradespace Analysis".

Bounds of the tradespace

TAT-C Tradespace Search Request (TSR – Setting Advanced Options)

Menu/Macro Buttons used to simplify TSR-build workflow

Tree View of Tradespace Search Request (TSR) sections

Dynamic View of Widgets used to Build/Portray currently-selected section of TSR Tree

Set by Simple Bounds   Load TSR   Save TSR   Run Tradespace Analysis



# Graphical User Interface

The screenshot displays a complex graphical user interface for managing tradespace analysis. It includes:

- Bounds of Interest:** A section for defining search parameters with input fields for Max. cost [\$], Max. number of observatories [#], Min. total time spent observing [s], Max. total time spent observing [s], Max. coverage time [s], Max. revisit time [s], Min. integrate time [s], Max. downlink latency [s], Max. spatial resolution [m], Min. altitude [m], and Max. altitude [m]. A "Set Advanced Options" button is also present.
- Progress through TAT-C Trade Space:** A progress bar showing "Percent Complete" at 50%, with a "Cancel Tradespace Analysis" button below it.
- Tree View of Tradespace:** A hierarchical tree view showing the structure of the tradespace search, including wxPythonDemo, Panel, Notebook, wxTreeCtrl, SearchCtrl, TextCtrl, statusbar, Frame, InspectionFrame, Toolbar, and Panel.
- Request (TSR) sections:** A large area labeled "Request (TSR) sections" containing several "Dynamic View of Widgets" sections for different parameters:
  - FS\_LEFT: Value 0.15, Set Digits 2, Set Format %f, %e, %g.
  - FS\_RIGHT: Value 1.4000e+000, Set Digits 4, Set Format %f, %e, %g.
  - FS\_CENTER: Value 0.01100, Set Digits 4, Set Format %f, %e, %g.
  - FS\_READONLY: Value 30.4, Set Digits 2, Set Format %f, %e, %g.Each section includes "Set Value", "Set Digits", "Set Format", and "Set Font" buttons, along with "Snap Ticks" checkboxes.
- Advanced Options:** A section for managing the R-build workflow, with buttons for Set Value, Set Digits, Set Format, and Set Font for FS\_LEFT, FS\_RIGHT, FS\_CENTER, and FS\_READONLY.
- Buttons at the bottom:** Set by Simple Bounds, Load TSR, Save TSR, and Run Tradespace Analysis.



# Graphical User Interface

**TAT-C Tradespace Analysis Results (Reports/Visualizations Browser)**

File Edit Draw Move Jobs Help

Menu/Macro Buttons used to simplify results browsing

Top-level Windows

- wxPythonDemo ("frame")
  - Panel ("panel")
    - + Notebook ("notebook")
  - Panel ("panel")
    - wxPythonDemoTree ("wxTreeCtrl")
    - SearchCtrl ("searchCtrl")
    - StaticText ("staticText")
    - TextCtrl ("text")
    - StatusBar ("status\_bar")
  - Frame ("frame")
  - Frame ("frame")
  - InspectionFrame ("frame")
    - Toolbar ("toolbar")
    - Panel ("panel")
  - Frame ("frame")

**Tree View of Tradespace Analysis Results to Browse**

```
pylab.axes([0, len(X)-1, 0, 1])
pylab.ylabel('Hour')
pylab.xlabel('Day')

# pylab.contour(x,y,z)
pylab.contourf(x,y,z)

else:
    fig=pylab.figure()
    ax = p3.Axes3D(fig)
    if sys.argv[1] == "1": ax.
    elif sys.argv[1] == "2": a
    elif sys.argv[1] == "3": a
    else: ax.contourf(x,y,z)
    ax.set_xlabel('Day')
    ax.set_ylabel('Hour')
    ax.set_zlabel('Hit Count')

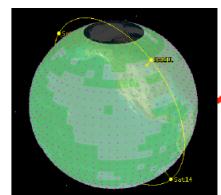
    go plot it!

pylab.show()
trainee@kingston:~/elen/analytics$
```

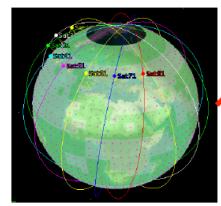
**Dynamic View of Reports/Visualizations for currently-selected section of TSR Tree**



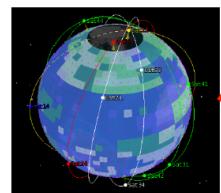
# Graphical User Interface – Proposed Outputs



Lat40\_fov15\_Alt400\_Inc60\_Planes1\_Sats1.cvaa  
Lat40\_fov15\_Alt400\_Inc60\_Planes1\_Sats3.cvaa



Lat40\_fov30\_Alt400\_Inc60\_Planes1\_Sats1.cvaa  
Lat40\_fov30\_Alt400\_Inc60\_Planes1\_Sats3.cvaa  
Lat40\_fov30\_Alt400\_Inc60\_Planes1\_Sats5.cvaa  
Lat40\_fov30\_Alt400\_Inc60\_Planes1\_Sats7.cvaa  
Lat40\_fov30\_Alt400\_Inc60\_Planes1\_Sats9.cvaa



Lat70\_fov15\_Planes6\_Sats2.cvaa  
Lat70\_fov15\_Planes4\_Sats4.cvaa  
Lat70\_fov15\_Planes4\_Sats3.cvaa  
Lat70\_fov15\_Planes4\_Sats2.cvaa  
Lat70\_fov15\_Planes4\_Sats1.cvaa  
Lat70\_fov15\_Planes3\_Sats4.cvaa  
Lat70\_fov15\_Planes2\_Sats8.cvaa  
Lat70\_fov15\_Planes2\_Sats6.cvaa  
Lat70\_fov15\_Planes2\_Sats4.cvaa  
Lat70\_fov15\_Planes2\_Sats2.cvaa  
Lat70\_fov15\_Planes1\_Sats16.cvaa

```
bcov 6.0
EpochTime: 12 Dec 2013 12:00:00.000000
NumberOfAssets: 6
0 Constellation/myWalker
1 Satellite/Sat1/Sensor/Sens1
2 Satellite/Sat2/Sensor/Sens2
3 Satellite/Sat3/Sensor/Sens3
4 Satellite/Sat4/Sensor/Sens4
```

```
PointNumber: 1
Lat: -8.31109167616347e-001
Lon: 1.74532925199433e-001
Alt: 0.000000000000000e+000
NumberOfAccesses: 15
5 1.12709670050281e+004 1.13330898171319e+004
4 1.24211387804788e+004 1.24767210766280e+004
2 7.24962572329515e+004 7.25624742138773e+004
3 7.36533545700632e+004 7.37006041267108e+004
5 9.68043718547335e+004 9.68725048141752e+004
```

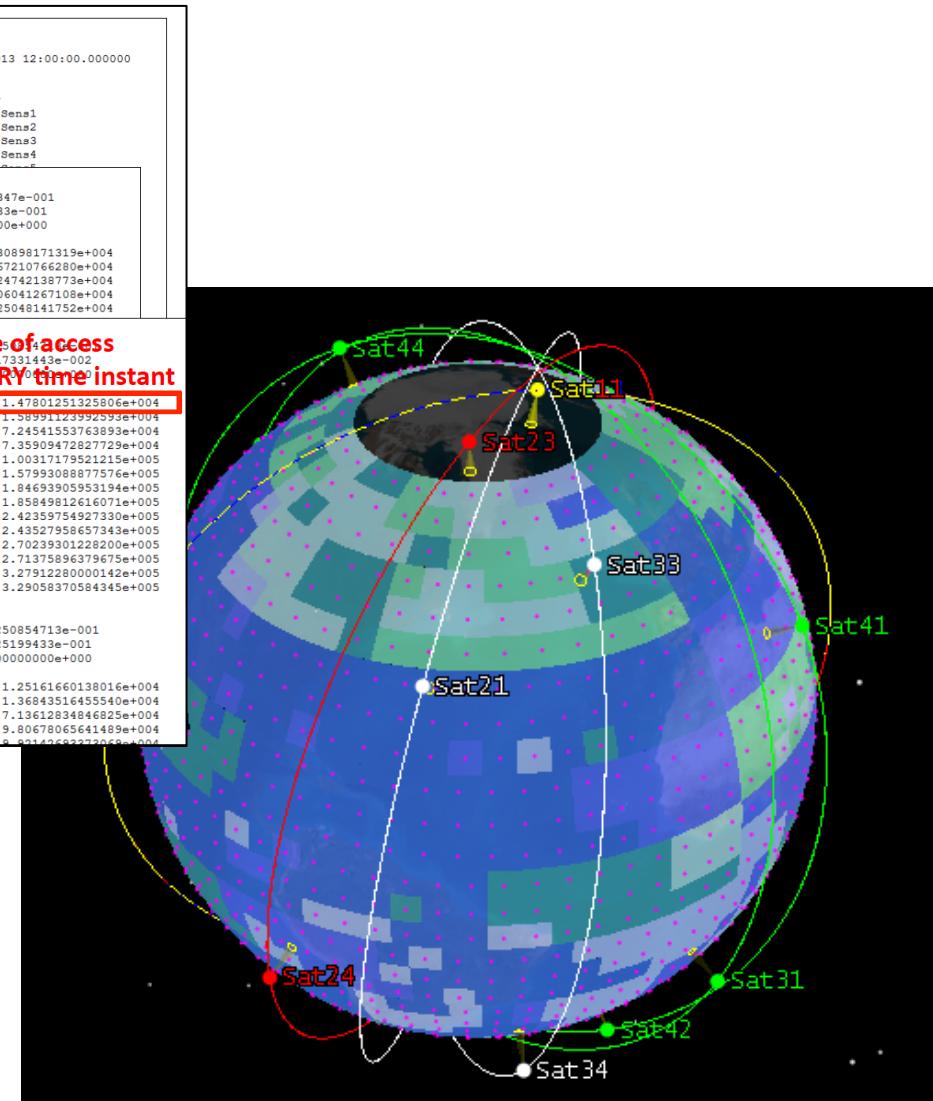
```
PointNumber: 2
Lat: 5.8116417331443e-002
Lon: 1.0024887656070e+005
Alt: 1.003317179521215e+005
NumberOfAccesses: 13
2 1.47136484262984e+004 1.47801251325806e+004
1 1.3886397798499e+004 1.38591123992399e+004
4 2.23916767419256e+004 7.24541553763893e+004
3 7.35490000596518e+004 7.3509472827299e+004
5 1.0024887656070e+005 1.003317179521215e+005
```

```
PointNumber: 3
Lat: 4.84650011861303e+004
Lon: 3.185788103852602e+005
Alt: 1.24238443669734e+005
NumberOfAccesses: 15
2 1.57924629637720e+005 1.57924629637720e+005
1 2.42338443669734e+005 1.84653905953194e+005
3 2.42338443669734e+005 1.85849812602e+005
5 2.43462066514163e+005 2.43527958657343e+005
2 2.70177550910218e+005 2.70239301228200e+005
1 2.71333472521181e+005 2.71375896379675e+005
1 3.278571182938763e+005 3.27912280000142e+005
3 3.29004393363166e+005 3.29058370584345e+005
```

```
PointNumber: 4
Lat: -7.47998250854713e+001
Lon: 1.74532925199433e-001
Alt: 0.000000000000000e+000
NumberOfAccesses: 13
4 1.24915905435807e+004 1.25161660138016e+004
3 1.36181690167946e+004 1.36843516455540e+004
2 1.2930264354796e+004 7.13612834846925e+004
5 9.80130904319586e+004 9.80678065641489e+004
1 9.81692045727172e+004 9.8214269272066e+004
```

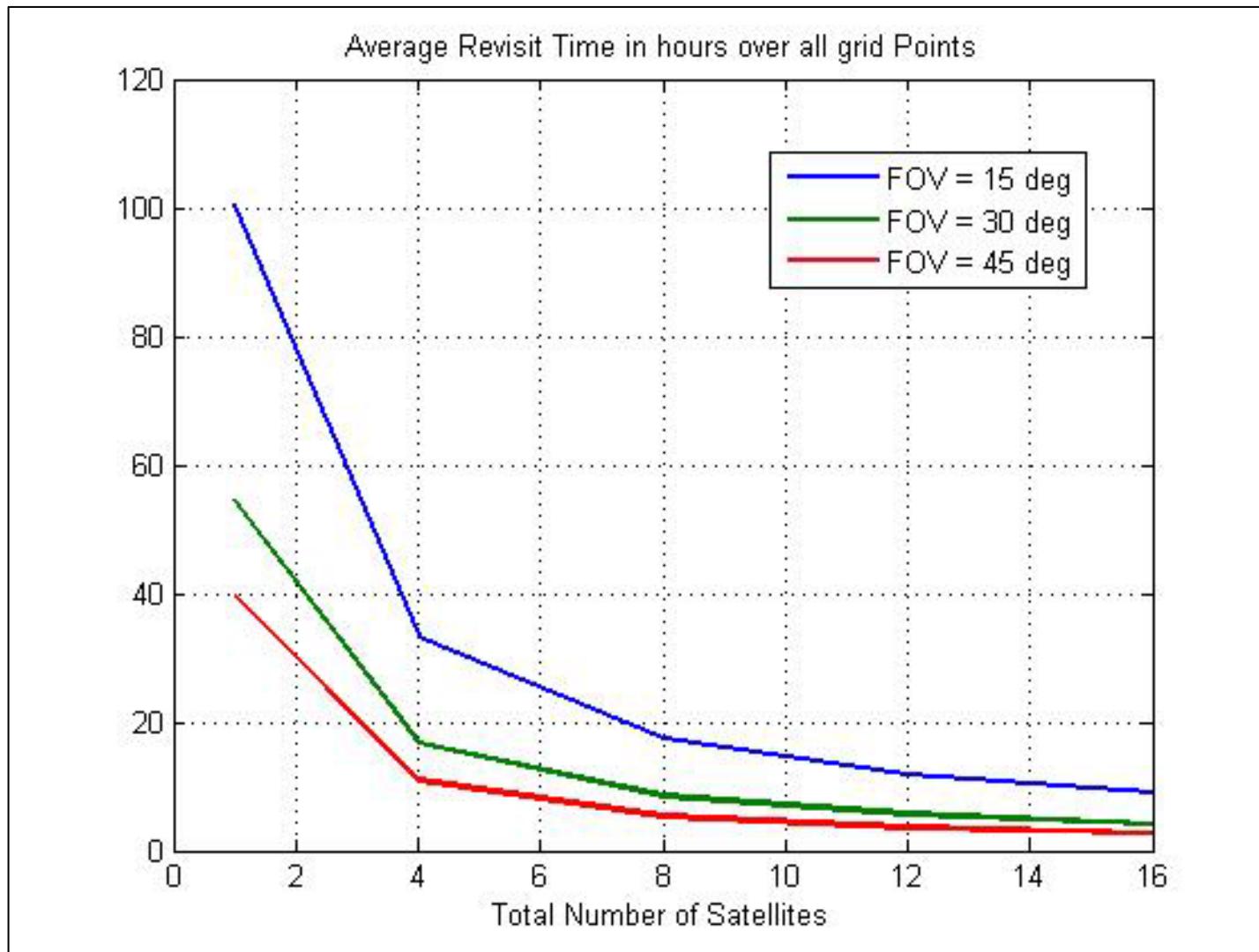
+ angle of access  
At EVERY time instant

## Constellation Orbit





# Graphical User Interface – Proposed Outputs



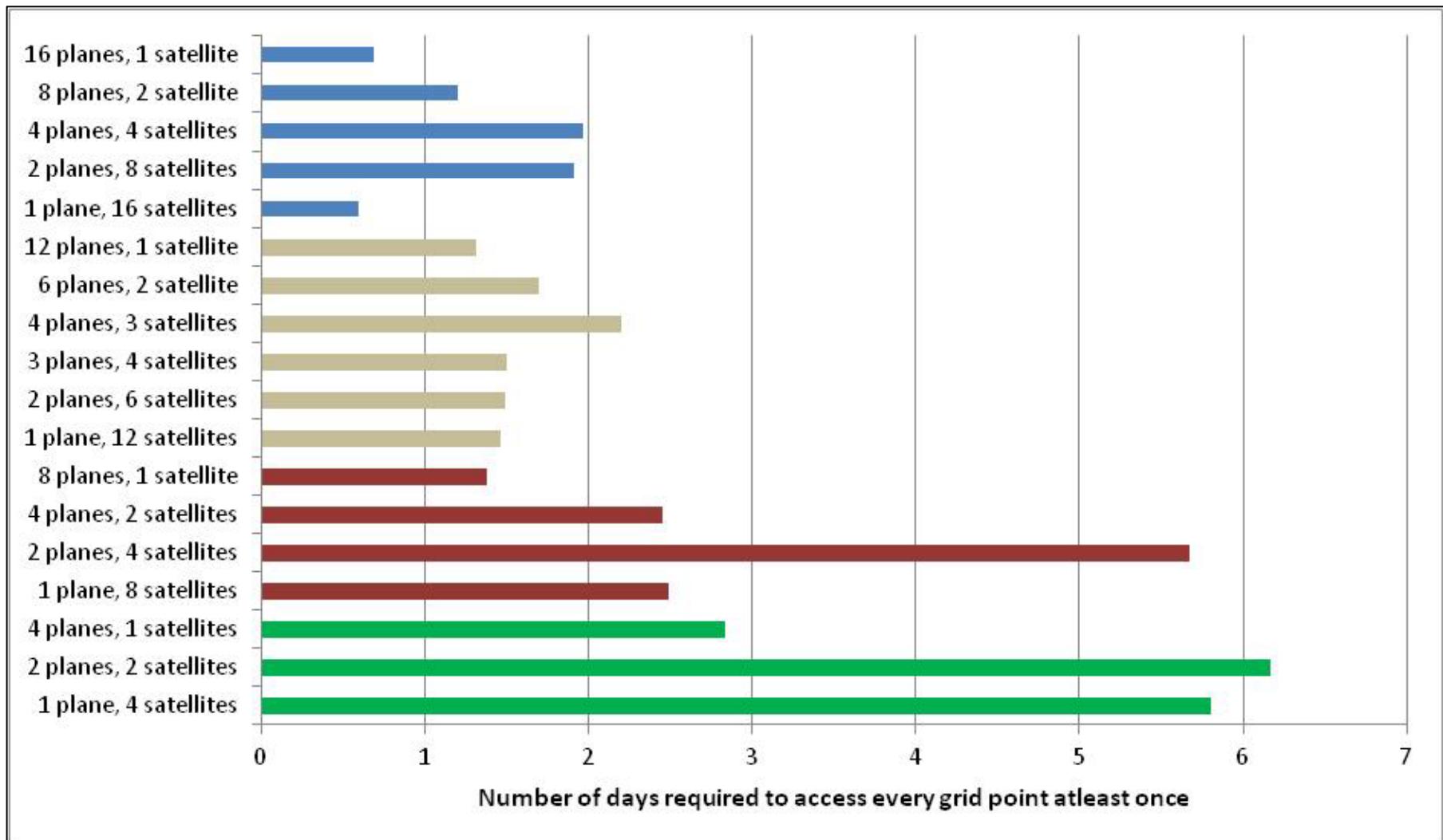


# Relevancy Scenario – The Sustainable Land Imaging (SLI) Program

- **Meet the coverage requirements**
  - Minimum composite tile size for a single “image” acquisition (90 kmx90 km)
  - Global land cover and some coastal waters equivalent to ~ 500 WRS2 scenes/day
- **Maintain trending and change sensitivity:**
  - Limited range of BDRF angles for both view and solar illumination: defines useable data within a sensor’s field of regard & local overpass times, e.g. - view zenith <30 degrees and local solar time within 30 minutes of 10 am
- **Minimum number of spectral bands for relevant products**
  - Some sensors may only provide a subset of Landsat-8 (SLI baseline) bands
  - “Less capable” satellites may contribute to the SLI community’s data
  - Assessing this allows the capability “holes” to be identified
- **Spatial resolutions**
  - Can the 30m equivalent multispectral band image be derived from the raw data?
- **Assess Acquisition frequency and latency for BDRF restrictions and Band combinations**
  - Community desires a greater re-visit/acquisition frequency



# Graphical User Interface – Proposed Outputs





# Summary and Future Plans

---

- Definition of External Requirements and Science Requirements
  - Interviews of Stakeholders
  - Definition of Science Products and Required Visualizations
  - GUI Storyboard and Command Line Interface mockups
- Implementation
  - Definition of TAT-C System Architecture and Overall Control Flow
  - All modules version 1 ready
  - *Expect Version Prototype demo in July 2016*
- Future Plans
  - Currently, some modules in Matlab and some in C++ (TSI in Python) => All computation-intensive modules in C++ (faster speed)
  - Refined user interface and TSI
  - Improved capabilities (Type of sensors, type of constellations, etc.)
  - Risk module and knowledge base
  - Use cases testing (starting with SLI)



---

---

# Thank you!



# Acronyms

---

---

- AIST: Advanced Information Systems Technology
- API: Application Programmer Interface
- AST: Architecture Study Team (for SLI)
- BAERI: Bay Area Environmental Research Institute
- CLARREO: Climate Absolute Radiance and Refractivity Observatory
- CLI: Command Line Interface
- COBRA: Complexity Based Risk Assessment Model
- CER: Cost Estimating Relationships
- C&R: Cost & Risk
- DSM: Distributed Spacecraft Mission
- ESTO: Earth Science Technology Office
- ED: Executive Driver
- FTA: Final Trade-space Analysis
- FOV: Field Of View
- GMAT: General Mission Analysis Tool
- KB: Knowledge Base
- MBSE: Model-Based System Engineering
- MIT: Massachusetts Institute of Technology
- NCIM: NASA Instrument Cost Model



# Acronyms

---

- O&C: Orbit & Coverage
- OSSE: Observing System Simulation Experiment
- R&D: Research & Development
- R&M: Reduction & Metrics
- SICD: Software Interface Control Document
- SLI: Sustainable Land Imaging
- SSCM: Small Satellite Cost Model
- STK: Systems Tool Kit (from AGI, Analytical Graphics, Inc.)
- TAT-C: Trade-space Analysis Tool for Constellations
- TSI: Trade-space Search Iterator
- TSR: Trade-space Search Request
- (G)UI: (Graphical) User Interface
- USCM: Unmanned Space Vehicle Cost Model
- USGS: United States Geological Survey
- VARG: Value-At-Risk-Gain
- VCRM: Value Centric Risk Management

# Generating DSM Architectures

Iterator Inputs

L  
oop

Loops to generate  
unique architectures:

*For every FOV in 4 equally distributed points in input FOV Range:* Make 4 equally distributed points in input Altitude Range. If special orbits, altitude is constrained.

*For every Altitude option among the 4 equally distributed points:* Calculate the swath on the ground for the given alt and FOV. Use swath to calculate the minimum and maximum number of satellites needed to achieve max and min revisit constraints. For computed range of satellite numbers, get 10 equally distributed options in between.

*For every total Sat Num option among the 10 equally distributed points:* Calculate all possible Walker constellation configs with # of planes  $\times$  sats/plane = total sats. Each uniform config is defined by the initial RAAN and TA of the constituent sats.

*For every inclination in 4 equally distributed points in input Inclination Range:* Constrain inclination inputs if there is a special orbits constraint. Incl depends on alt option, if so.

*For every RAAN-TA combination for a given inc, sat number, alt, FOV (unique architecture):* For each sat, Compute the initial Keplerian elems. If the sat, GS and payload combination is unique, store all their specs as a unique JSON file under the User/Mono/ folder. For the whole DSM, Store the mapping of the unique sats that make up the DSM (pointer to Mono/ location) as JSON files under a unique folder for this arch in User/DSMs/ folder.

Then, create JSON file with information for each DSM for the cost/risk module (e.g., orbital params, maintenance delta-V)