Software Solutions for Distributed Autonomous Multi-Functional Robotics in Space

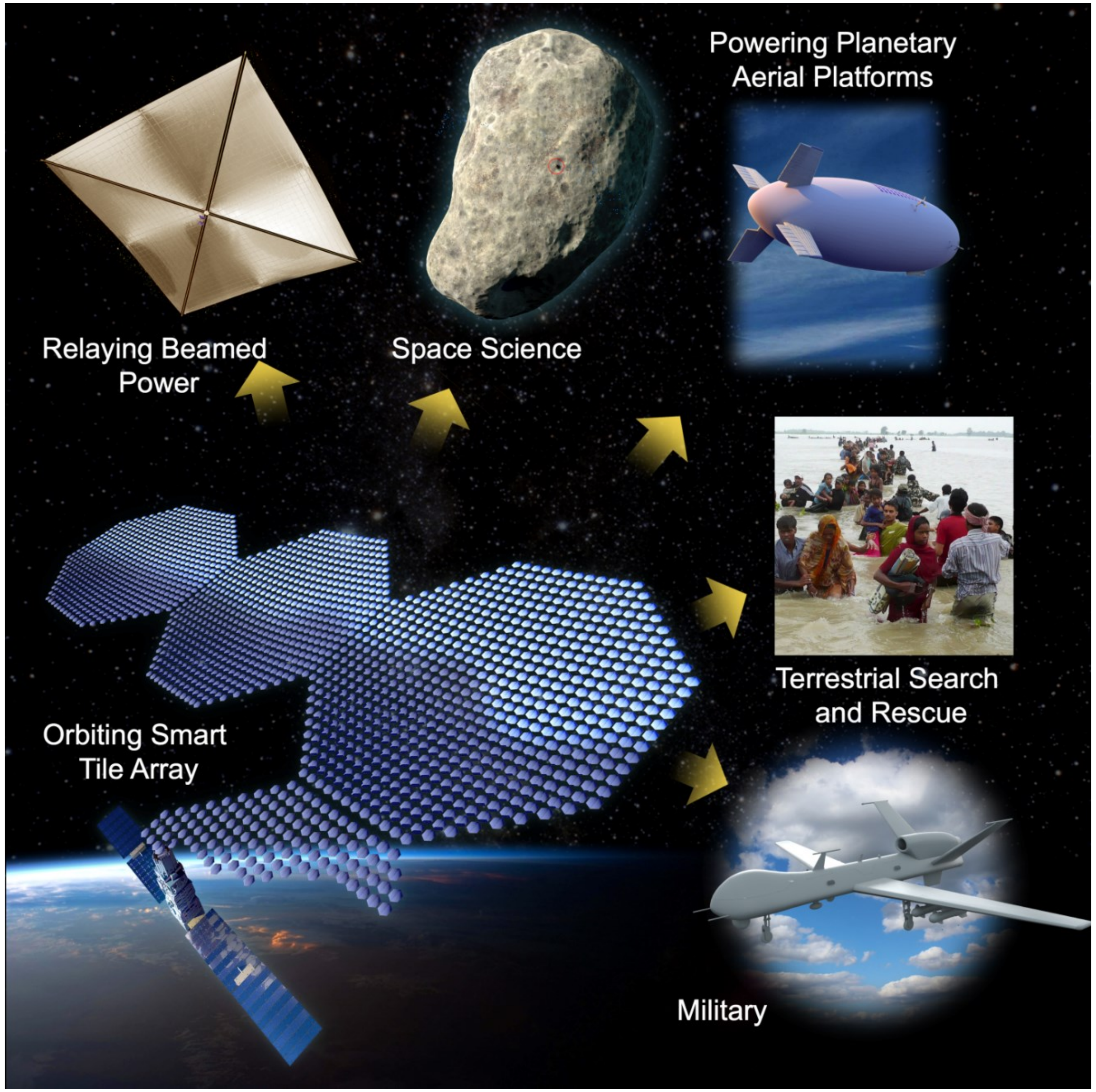
James Edmondson

Carnegie Mellon University

Software Engineering Institute

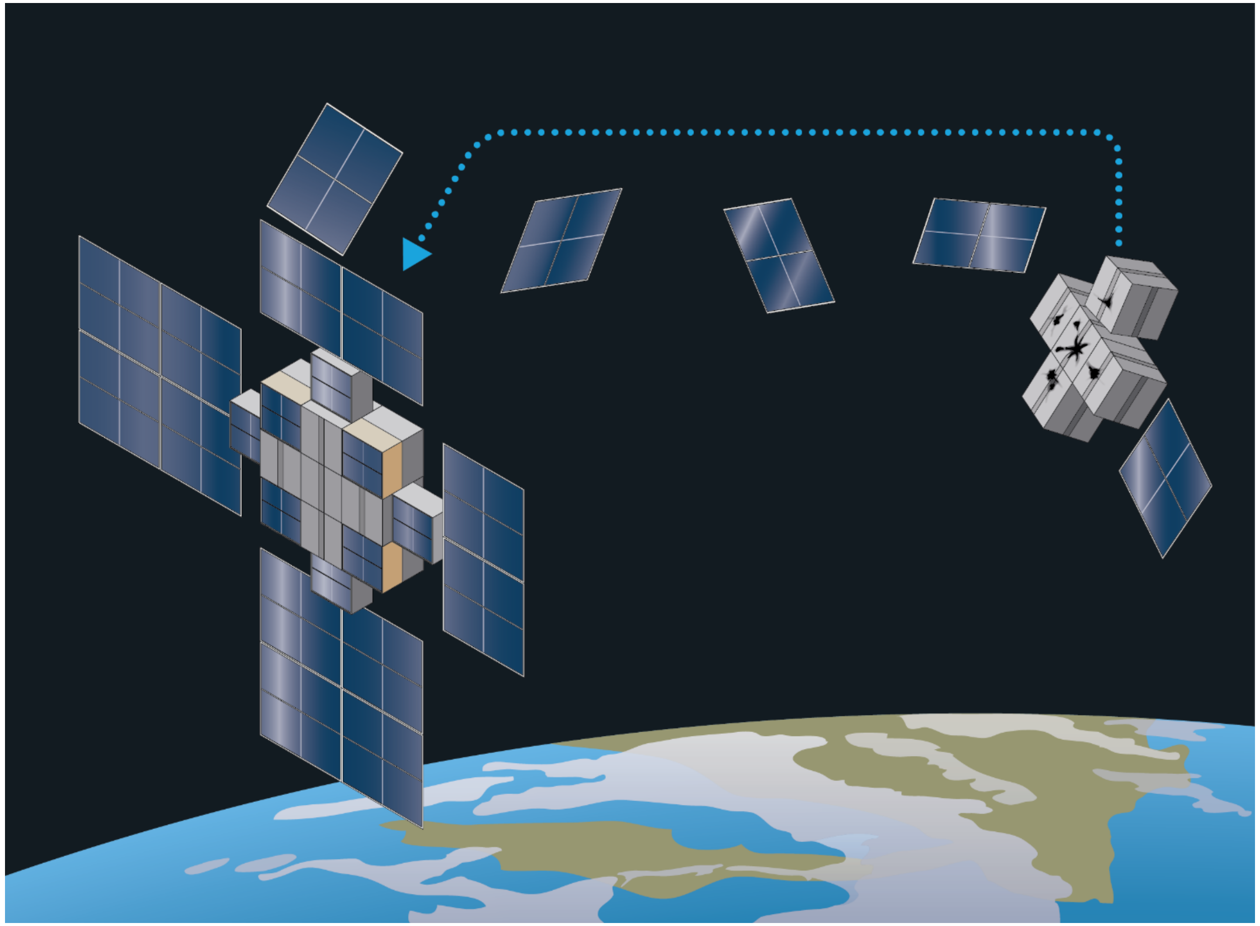
About the Multi-planetary Smart Tile

- The Smart Tile is a **multi-functional tile** that is intended for **energy gathering and transfer** throughout the solar system
- The Smart Tile **operates in space and** on **terrestrial surfaces**
- The Smart Tile is intended to be **deployed in large numbers**
- Smart Tiles **must collaborate** to accomplish most missions



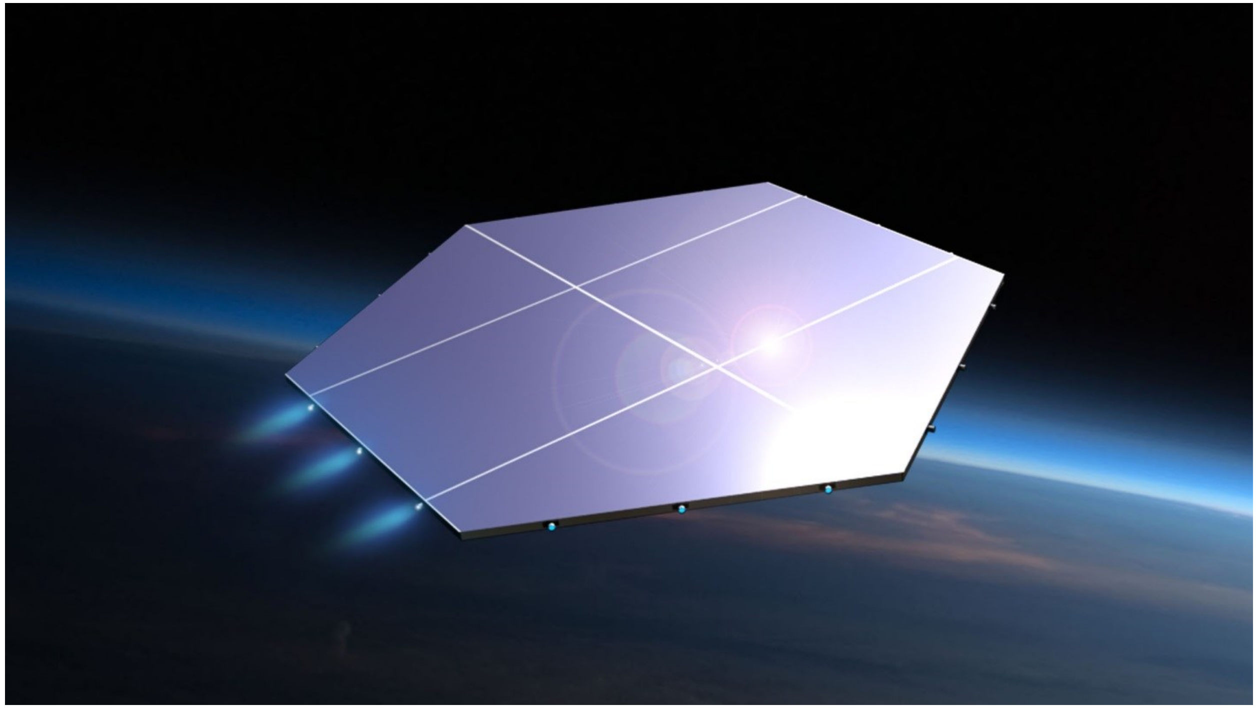
Why Distributed Autonomy?

- **Messages from earth**-based stations will **take minutes** to arrive and are often expensive (e.g., Iridium satellite costs)
- **Managing thousands of tiles** from minutes away **may not be possible**
- **Significant delays** in earth-sourced commands can **cause:**
  - **Loss of tile**
  - **Failure of mission**
  - Worst case **loss of life**, if tile is supporting local human operation
- Making **autonomous tiles** that **collaborate** results in **near-instantaneous** command execution**, more capabilities** for human operators and **better chance of mission success**

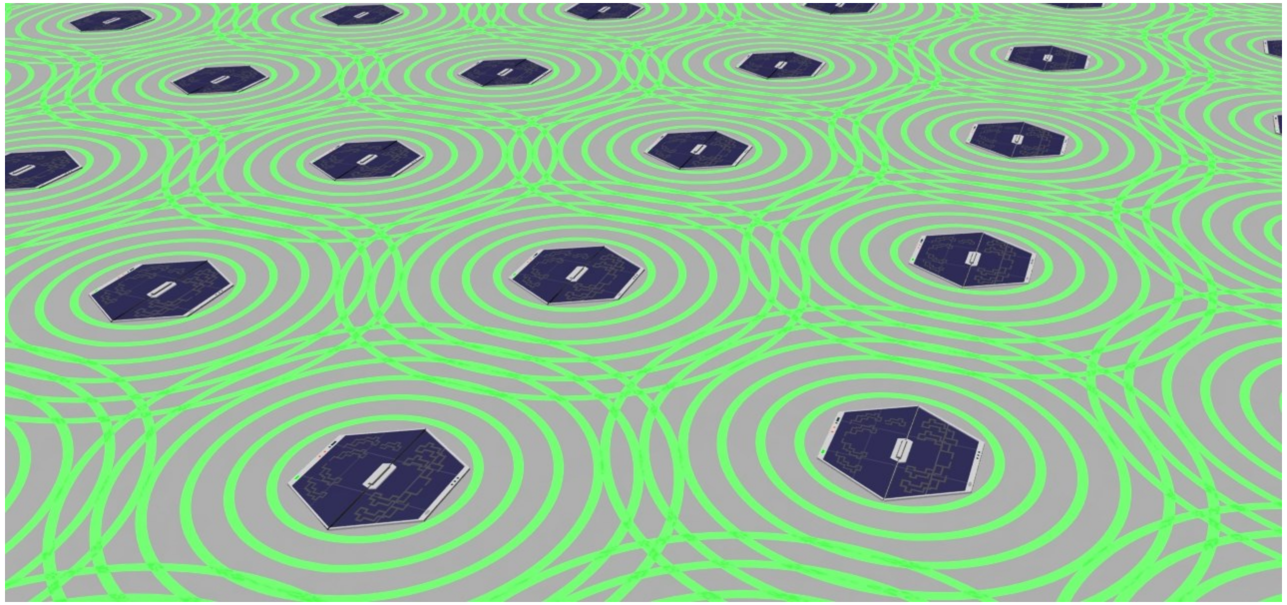


Space-Based Missions

- The Smart **Tile forms communities virtually and physically**
- **Virtual communities accomplish computational tasks** as a group
  - Involves **task decomposition, analysis, assignment and performance**
  - **Complex computation**al task
  - **Advanced problem solving**
- **Physical communities accomplish spatial tasks**
  - Energy **sharing, harvesting**
  - Providing **rigidity, structure and formation**
  - **Powering** other structures and **equipment**
  - **Localization** for navigation
  - **Sensing**
  - **Defense/shielding**



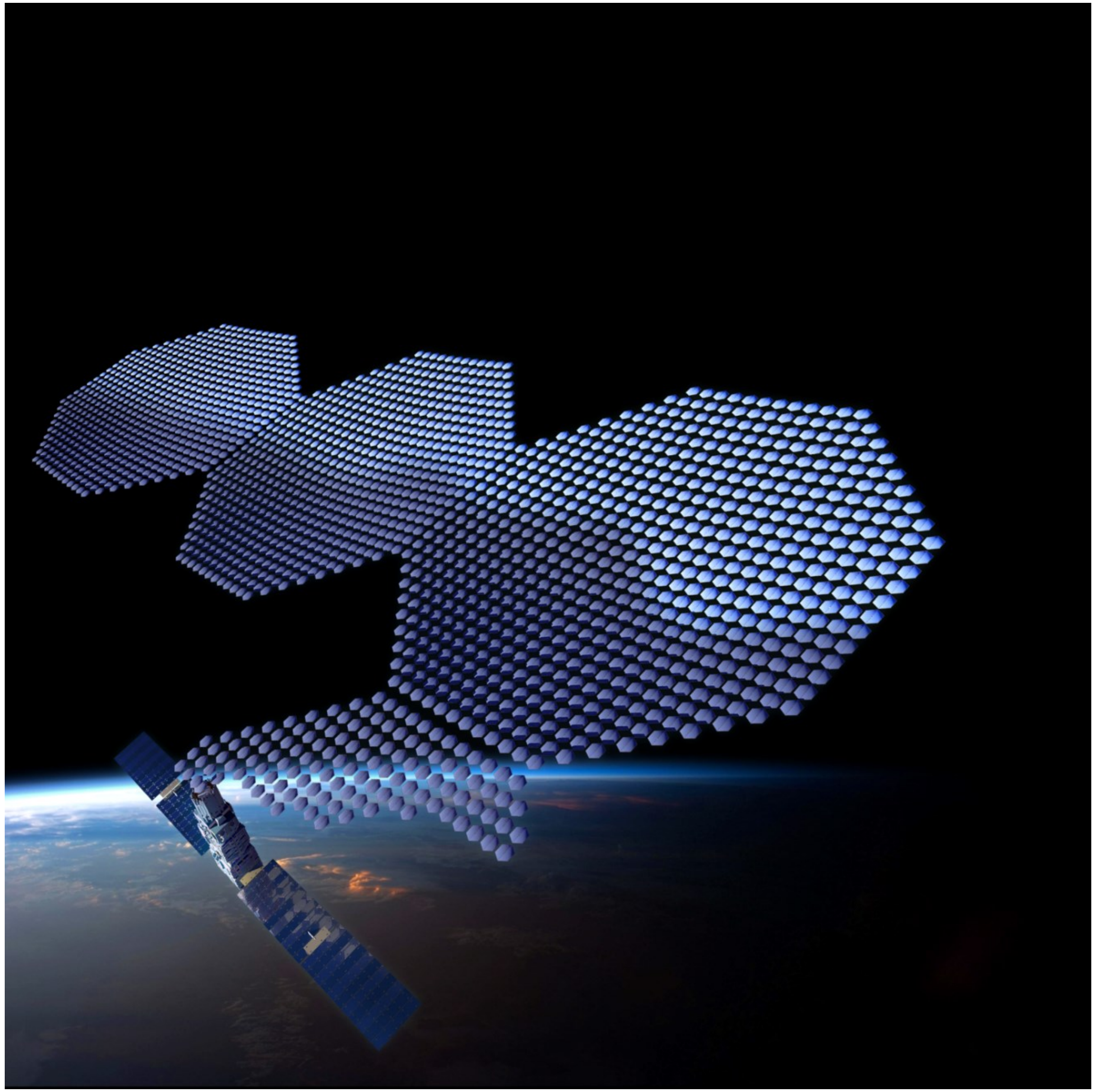
Common Challenges



- **C1: Latency**
- **C2: Communication loss**
- **C3: Common communication protocols** – how do the tiles share information
- **C4: Knowledge and Reasoning** – how do the tiles share a common understanding of a problem or mission?
- **C5: Task assignment**: which tile should do what?
- **C6: Trust and privacy**: for tile intent, for users, etc.
- **C7: Scalability**
  - Need to scale to thousands of participants over appropriate networking mediums
  - Need to filter messages over earth-based satellite feeds such as Iridium to limit expensive bandwidth consumption

Physical Challenges

- **C8: Collision avoidance** – tiles have the potential to move at high speed and collisions can damage the tiles
- **C9: Localization** – moving to the right location is important for all space-based missions. GPS does not exist throughout the solar system. Even on Earth, GPS is too coarse.
- **C10: Extensible Platforms** – tiles may be varying sizes with different capabilities. Solutions must support arbitrary platforms.

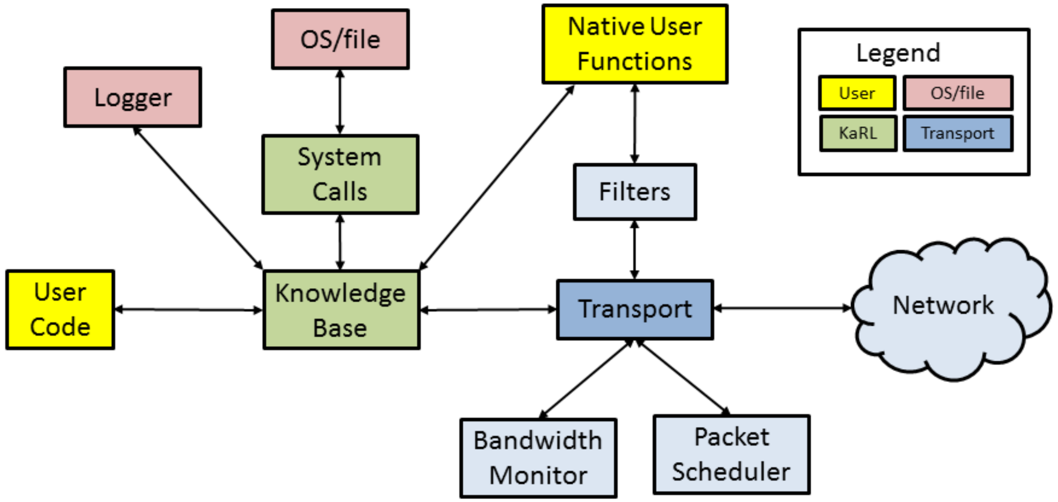


CMU Middleware Development

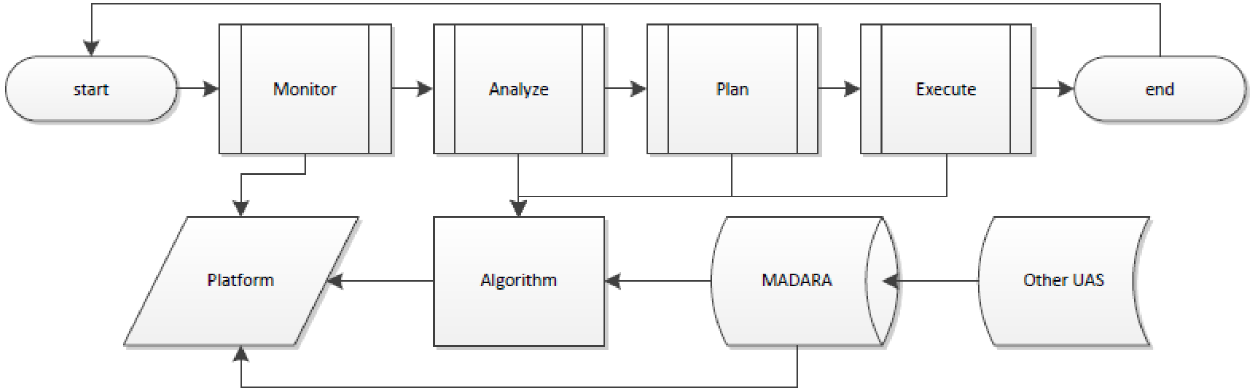
MADARA

- **Loss and latency-resistant** knowledge transfer between agents
- **Scalable knowledge and reasoning services**
- **Threading, control, networking**

http://madara.sourceforge.net



http://jredmondson.github.io/gams/

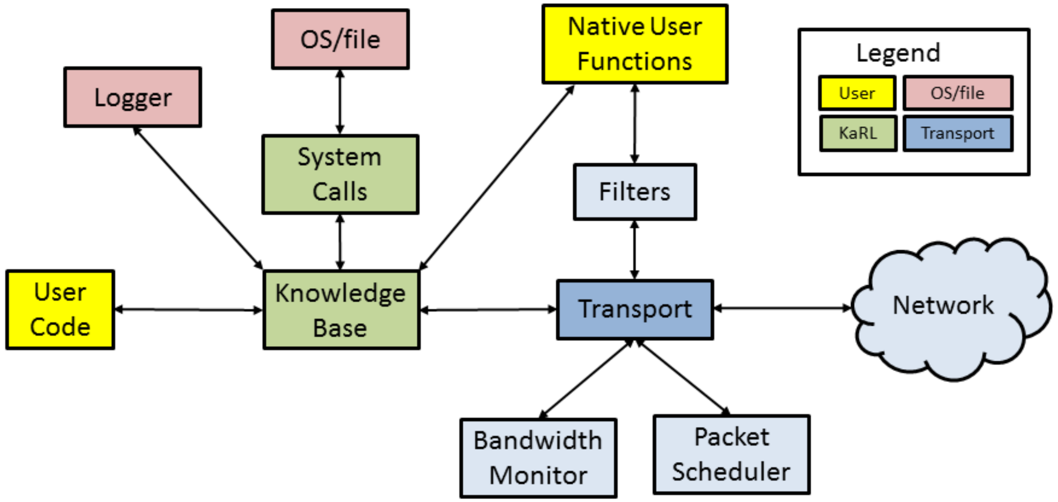


GAMS

- Provides structured reasoning framework for **common mission understanding**
- **Provides predictable execution**
- **Extensible algorithms**
- **Extensible platforms**
- **Focuses on group missions**

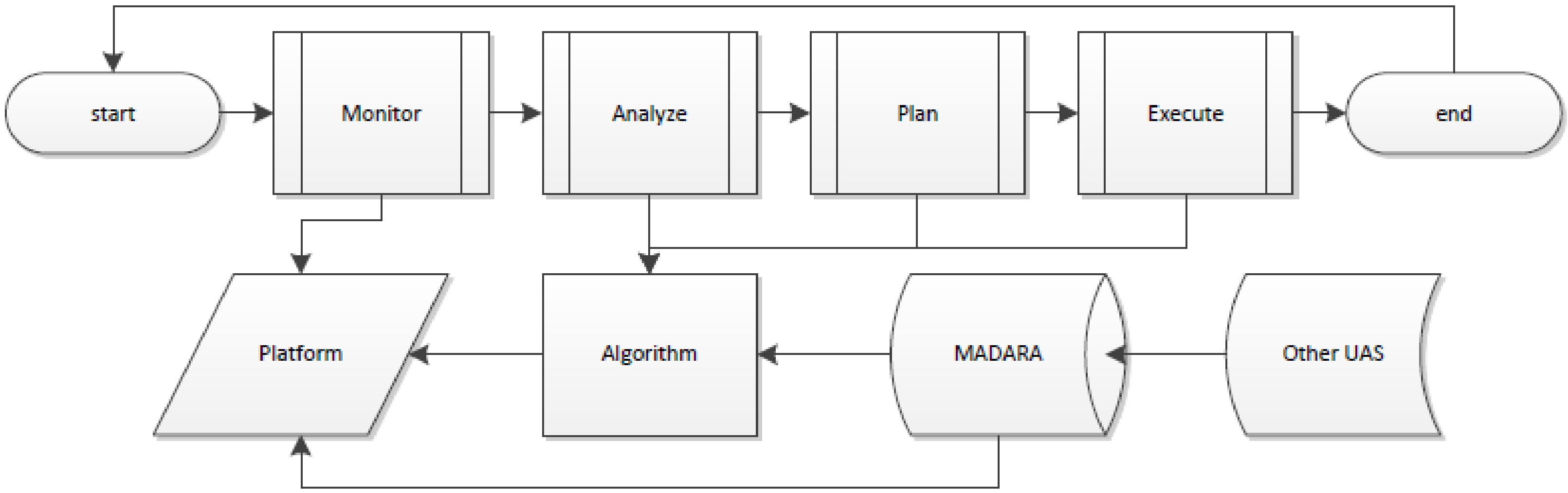
How MADARA works

- **Values** are **mapped to variable names** in a knowledge base (C4)
- **Changes to knowledge** are **aggregated** and **sent over network** (C3)
- **Consistency checks** are implemented via **Lamport clocks** per variable to allow for resends over unreliable transports
- **Designed** environments **for** variable **latency and loss** (C1, C2)
- Extensible filter system for **bandwidth and deadline shaping** (C7), **encryption and security protocols** (C6)



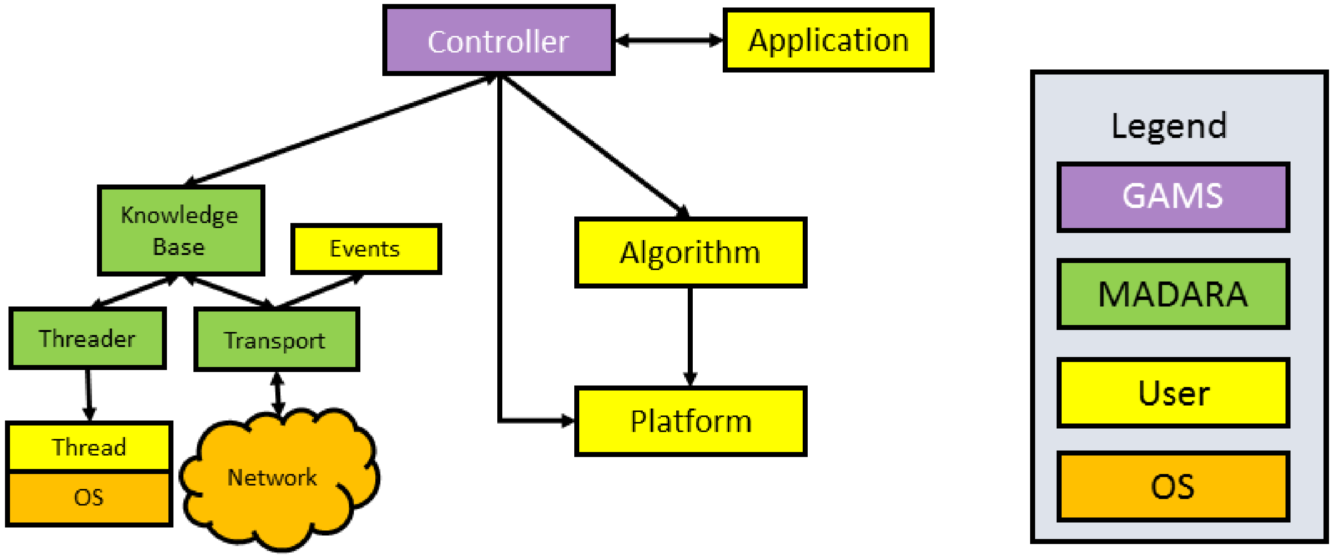
How GAMS works

- **Uses MADARA** for knowledge and reasoning (C1-4, C6-7)
- **Localization and heading/vector primitives** are provided per agent (C9)
  - Maps to any 3D Pose and Reference Frame, including translations
- Provides **algorithms** and an interface for **distributed tasks** (C5)
  - Currently implements **formation**, patrols, group defense, **coverage**, protection, etc.
- Provides **platforms** and **interface** for new platforms (C10)
- Provides **consistent MAPE-K** execution model (C4, C6)



How GAMS and MADARA fit together

- GAMS is built directly atop MADARA
- GAMS algorithms and platforms have access to all MADARA features
  - Threading
  - Networking
  - Knowledge
  - Timing and control



How scalable is this middleware? (C7)

- For **ARM processors,** this can scale to **thousands** of agents and potentially **tens of thousands, depending on send hertz** per agent
- **Following experiments** performed on a 20 node **ODROID** cluster of **ARM processors** with wired and wireless connections, **1 KB knowledge packet** (roughly 5x average packet)

	Min Hz	Max Hz	Avg Hz
Per Node Publish	500.00	517.30	507.46
Per Node Receive	432.00	490.00	468.12
Per Node Total Receive	8,820.00	9,082.00	**8,921.38**
Total Throughput			133,868.00
Node Publish	5.00	35.00	31.00
Per Node Total Receive	100.00	700.00	**620.00**
Total Throughput			12,400.00
Estimated scaling	1764	240	**280**

If we publish knowledge at **1hz,** even low powered ARM processors could scale to **8,800+** autonomous **tiles**

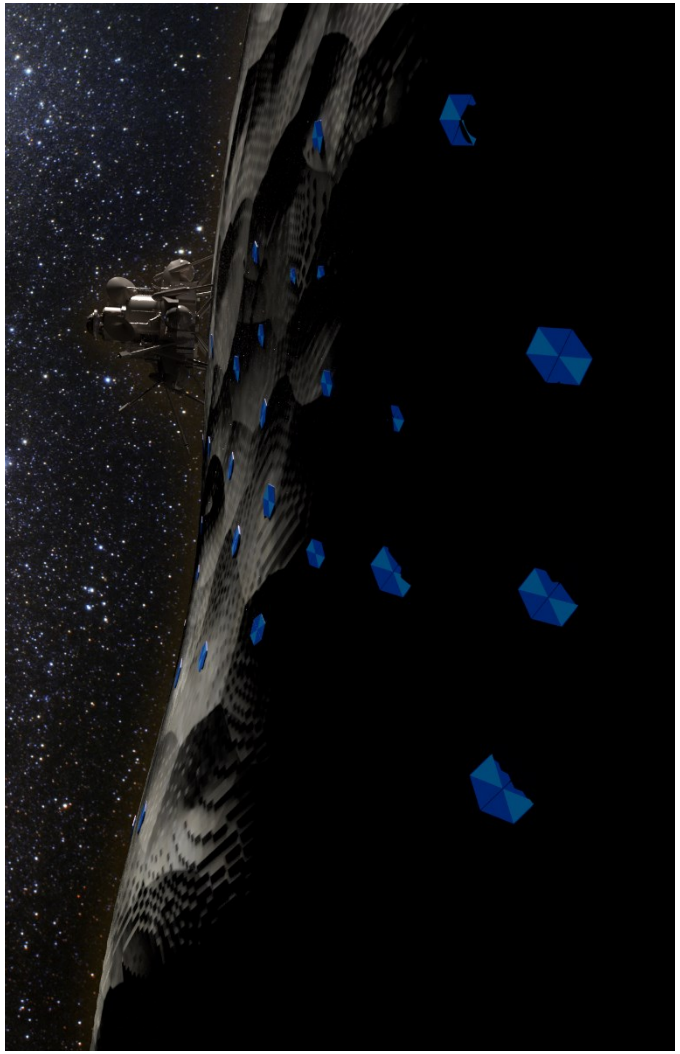
What kind of distributed AI is possible in space?

Spatial intelligences

- **Collaborative formations** for energy harvesting
- Dynamic **relays** for energy beaming/transfer
- Convoy **shielding** through asteroid belts and debris fields
- Reconfiguring to **optimal sensing** positions **around astral bodies**
  - **Localization** aids, **proximity** sensors, sensing **experiments**, etc.

Computational intelligences

- **Big data** analytics for science experiments
- **Weather forecasting** on Mars
- **Personal assistants** to space travelers
- **Deep learning**

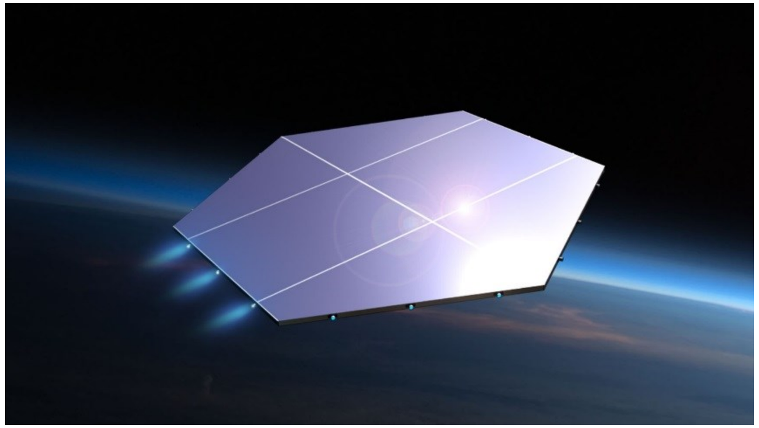


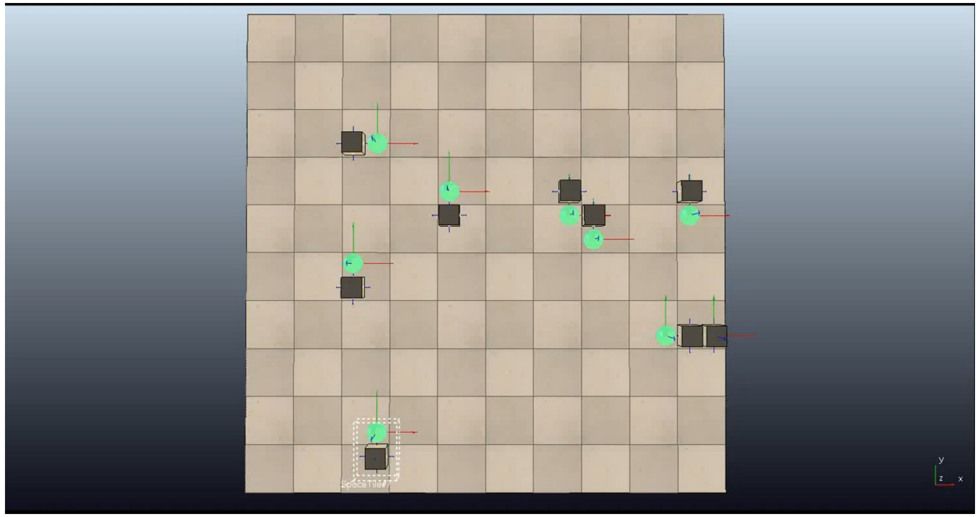
How do you make artificial intelligences for tiles?

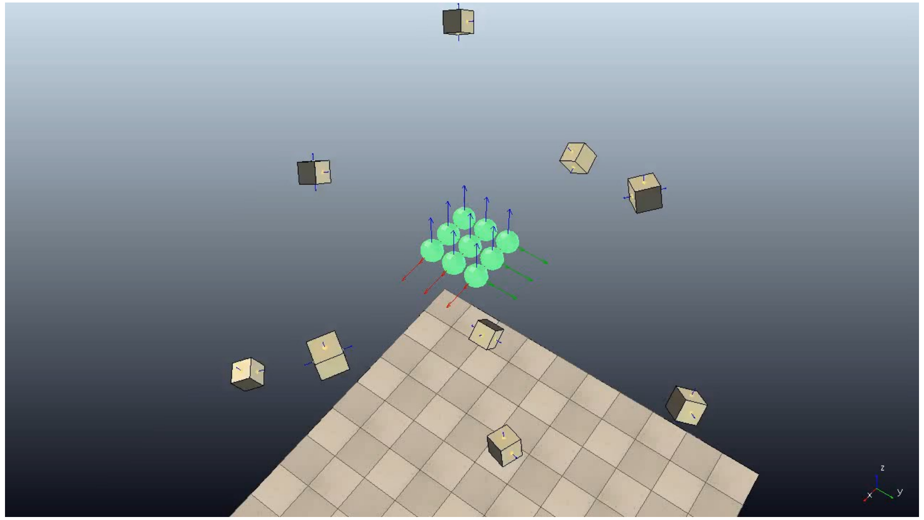
- Similarly to how we do AI for unmanned autonomous systems
- With GAMS
  - **Extend platforms** to include tile functionality
  - **Implement algorithms** to be executed within MAPE-K controller
- With MADARA
  - **Access distributed knowledge** that is **useful to** the collaborative AI **mission**
  - Rely on the transport layer to distribute knowledge and **reassemble global picture**
- Use an appropriate **model of computation**
  - **Synchronous models** of computation help collaborative agents **maintain distances, formations and ideal synergies**. They are **easier to formally verify**.
  - **Asynchronous models** of computation are more **reactive**, more **tolerant of latency and packet loss**

How might the tiles move in space?

- Several configurations:
  - Unidirectional thrust
  - Omnidirectional thrust
  - Wheeled (walks along other tiles)
- Algorithms for structured formations need to take into account the type of movement possible for the tile







What is still needed for distributed artificial intelligence with the Smart Tiles?

- Once the **prototypes** for smart tiles are **completed**, we can **begin work** on the software **architecture** and **verification** of Smart Tile components
- **Platform implementations** for the prototypes
- **Algorithm implementations** for all mission types
- **Verification of the platforms** via software model checking
- **Verification of the algorithms** via statistical and software model checking

Further reading

MADARA: http://madara.sourceforge.net

GAMS: http://jredmondson.github.io/gams/

DART: http://cps-sei.github.io/dart/index.html