ROSRV:
Runtime Verification for Robots

Jeff Huang, Cansu Erdogan, Yi Zhang, Brandon Moore,
Qingzhou Luo, Aravind Sundaresan and Grigore Rosu

University of Illinois at Urbana-Champaign
SRI International, Menlo Park, CA

Robot Operating System

- Meta-operating system for robot software development
  - Not a traditional operating system:

    No scheduling, process management

- Message passing interface based on a graph architecture
  - ROS processes (a.k.a. *nodes*)
    - Receive,
    - Post,
    - Multiplex messages

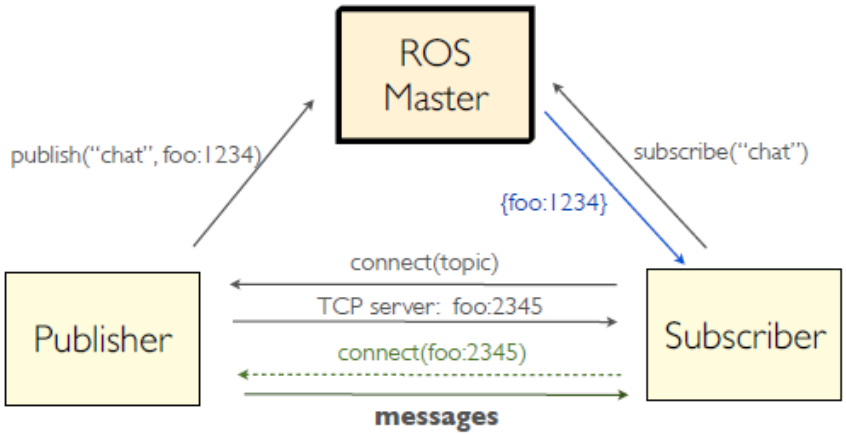


Fig. 1. ROS Communication Architecture

Motivation

Problem

- ROS is highly dynamic and distributed (and popular!)
  - No control over what nodes do
  - Lacks security protection mechanism

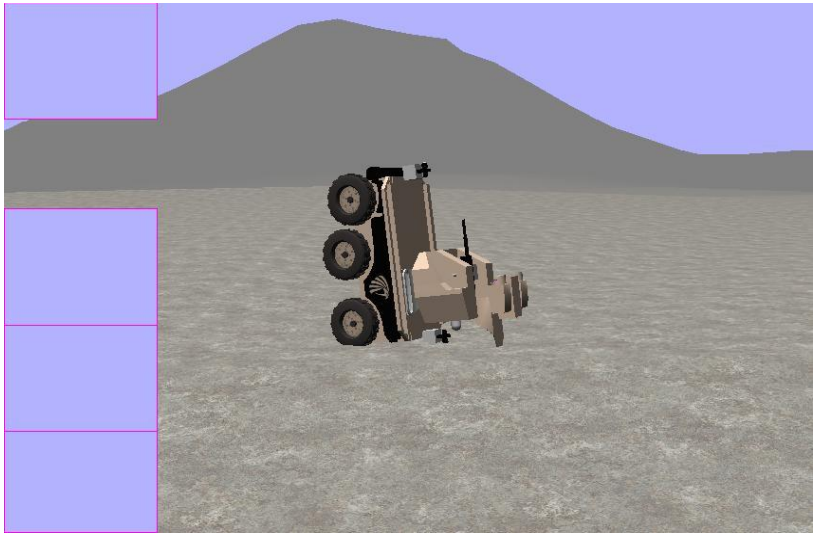
Solution

- Monitoring safety properties at runtime
- Enforcing security policies at runtime

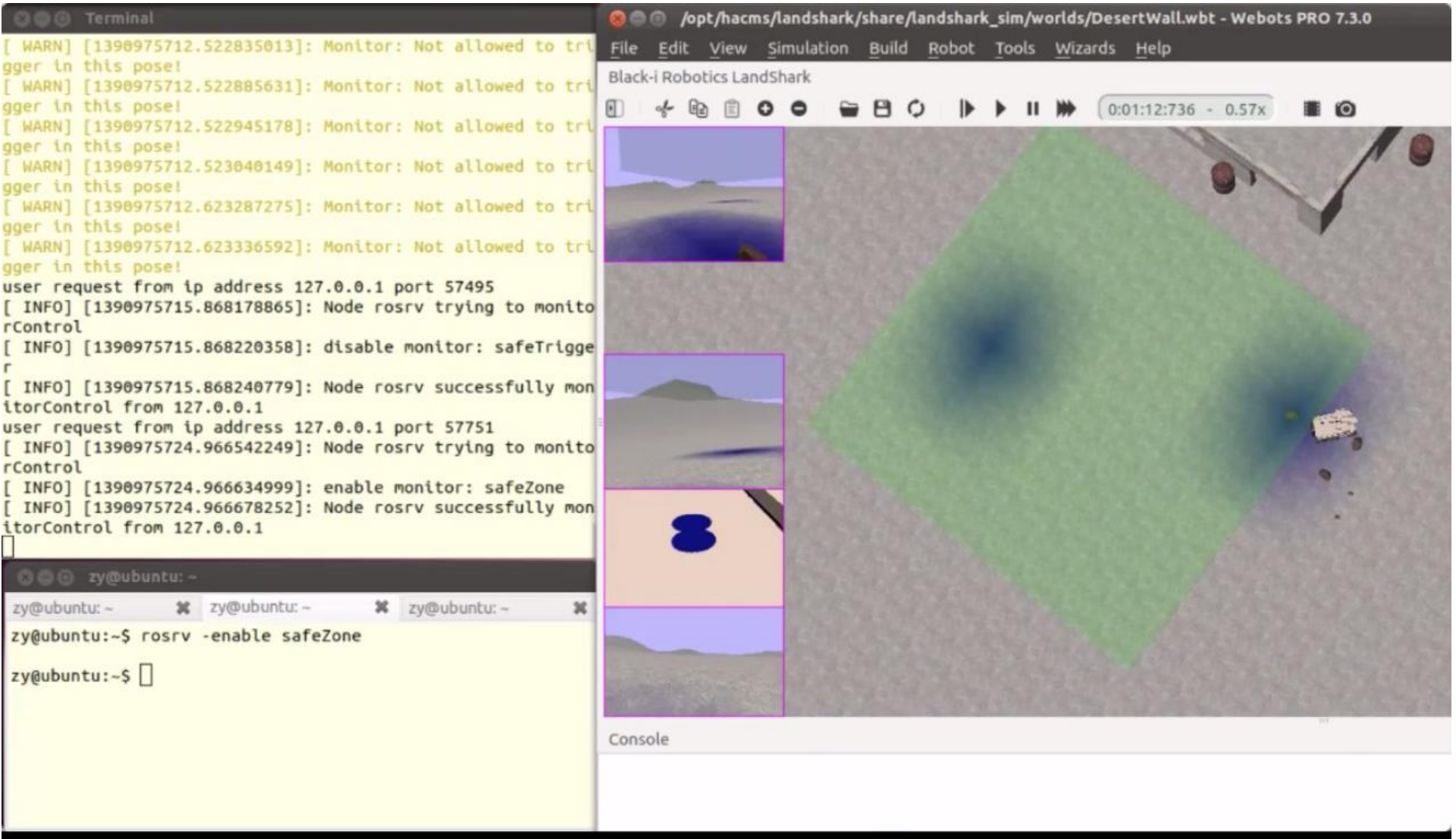
LandShark

Shoots itself

Flips over



Stays in safe zone



ROS…

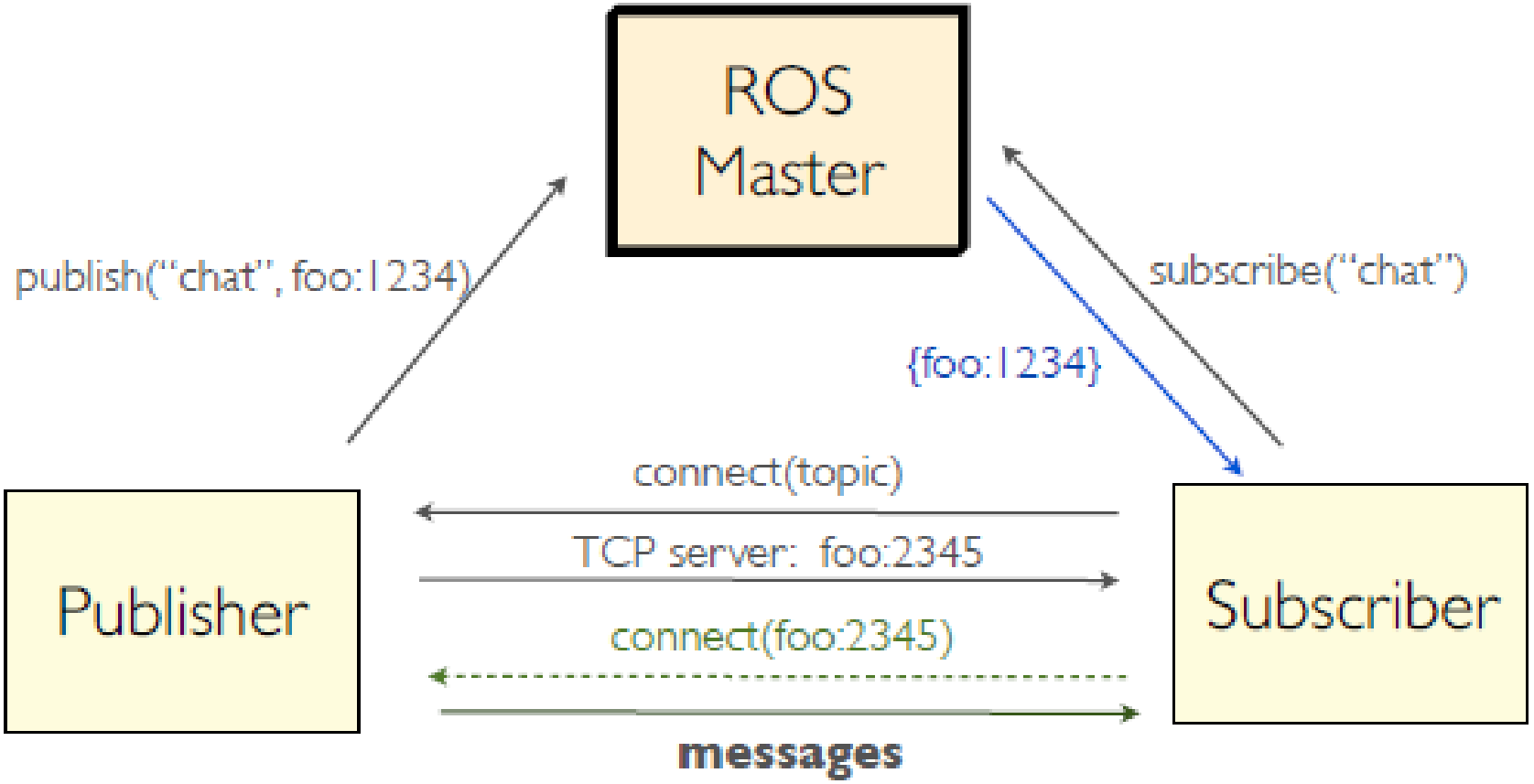
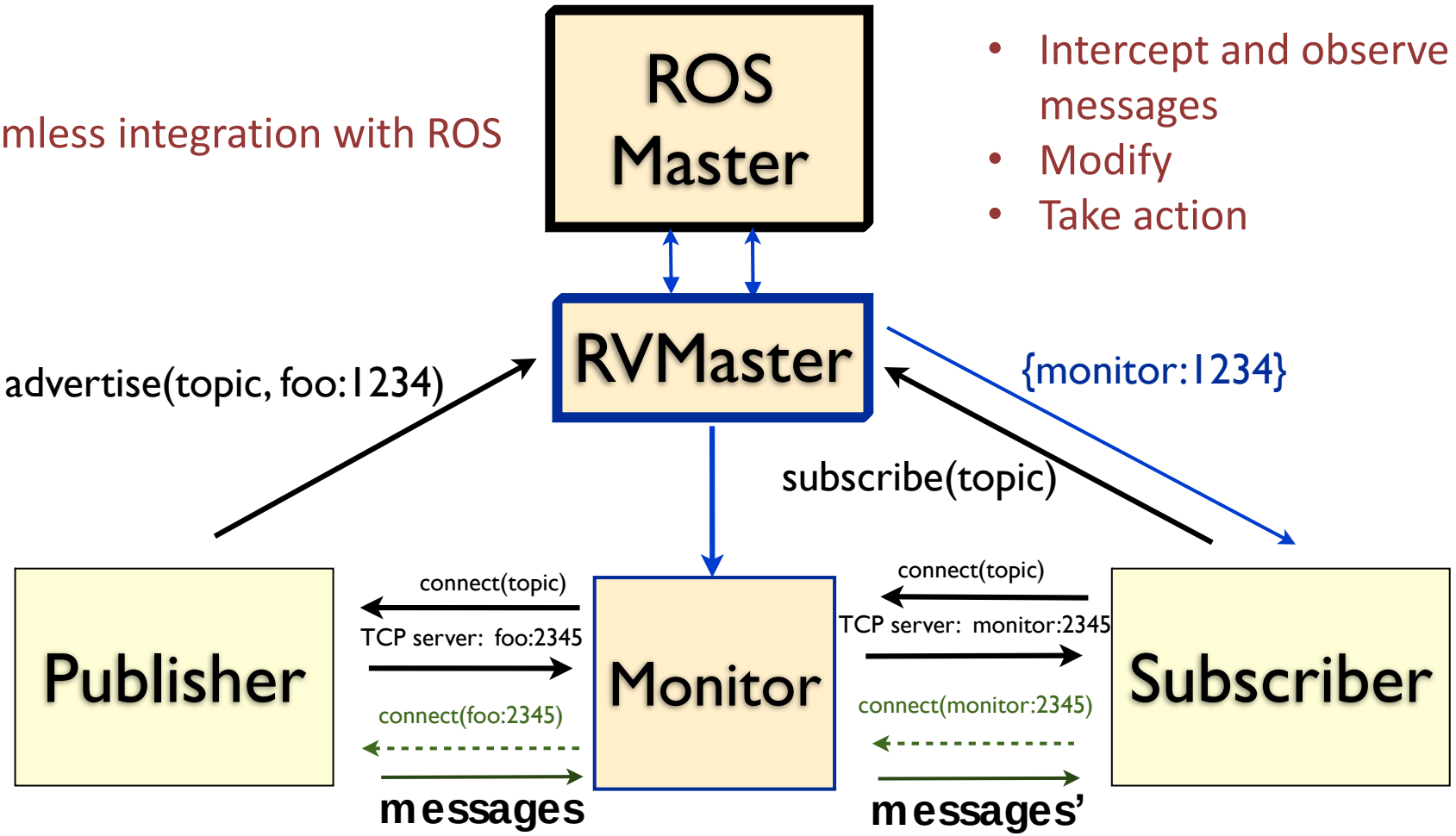


Fig. 1. ROS Communication Architecture

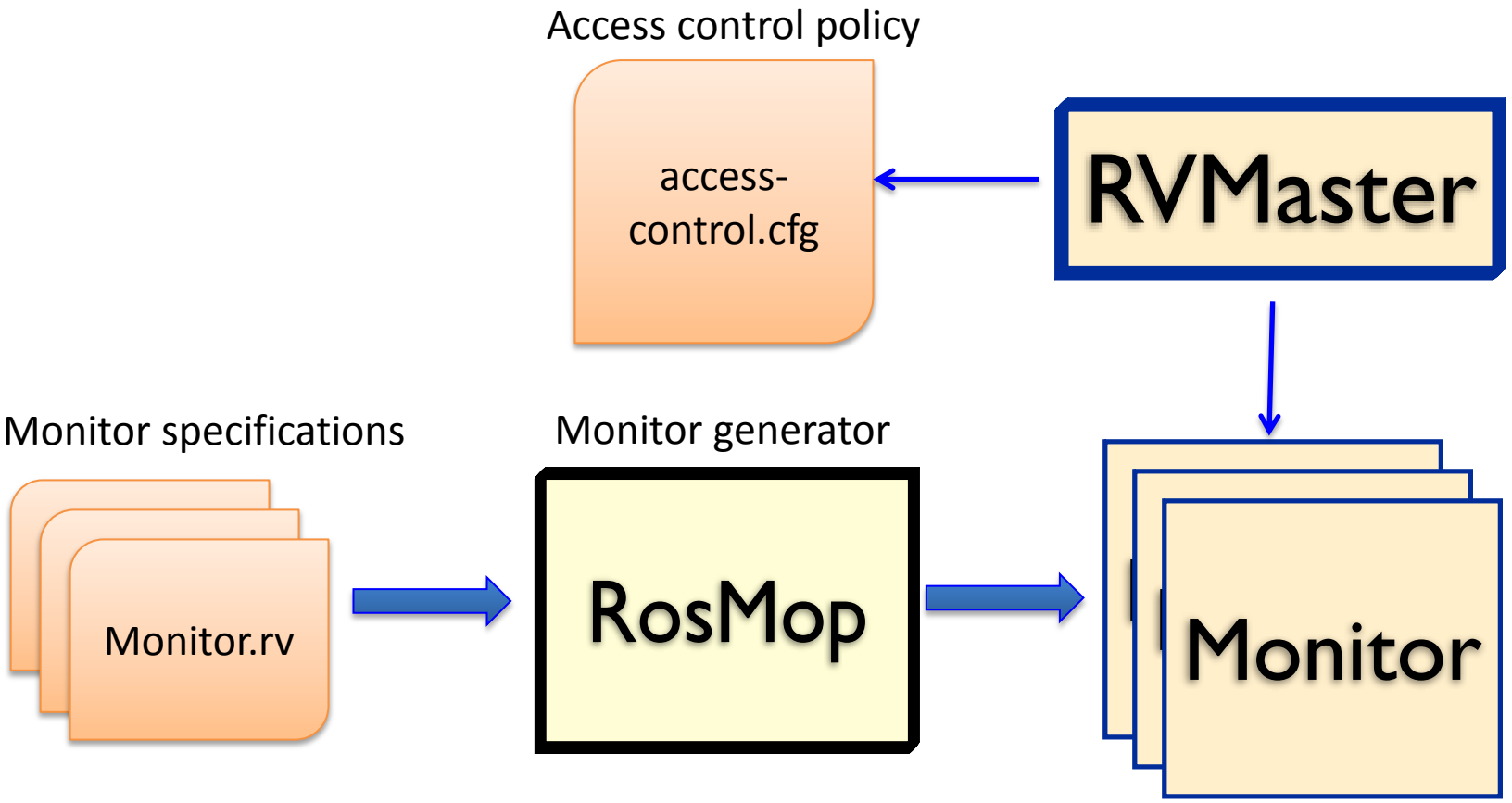
… vs. ROSRV

Seamless integration with ROS

- Intercept and observe messages
- Modify
- Take action



System Overview



Monitor Specifications

```
safeBehavior(){
  bool isSafeTrigger = false;
  event checkPosition(string N, double P)
    /landshark/joint_states sensor_msgs/JointState
    '{name[1]:N, position[1]:P}' {
        ...
        //check gun position
        ...
    }
  event safeTrigger() /landshark_control/trigger
    landshark_msgs/PaintballTrigger '{}' {
        ...
        //drop trigger message if not safe
        ...
    }

  ...
}
```

Access Control Policy

```
[Groups]
localhost = 127.0.0.1
certikos = ip1 ip2 ip3 ip4
ocu = ip5 ip6 ip7 ip8
[Nodes]
default=localhost
/landshark_radar=certikos
```

```
[Publishers]
default=localhost certikos
/landshark_control/trigger= ocu
[Subscribers]
default = localhost certikos
/landshark/gps = ocu
```

```
[Commands]
# Commands: full access
getSystemState = localhost certikos ocu
# Commands: limited access
lookupNode = localhost certikos
# Commands: local access only
shutdown = localhost
```

shutdown command cannot be sent by any node other than **localhost**

Future Work

- Formal verification
  - Monitors are correct
  - Nodes cannot bypass monitoring
  - No irrelevant events yield bogus errors
- Scalability
  - Decentralized approach

Conclusion

- Safer and more secure robots
- Automatically generated monitors
- User defined formal safety and security specifications

Thank you!

Questions?

Check our demo at:

http://fsl.cs.illinois.edu/ROSRV