Topics

- Overview
- Changes
- Request for help
- Discussion, next steps
Implementation model

ASA 1

ASA 2

ASA 3

ASA 4

API

Function library

Unicast sockets

GRASP core

Internal Functions

Discovery logic

Discovery multicast handler

Caches: (Discovery, Objectives, Flooded objectives, Sessions)

Flood multicast handler

Multicast sockets

IPC
Important data structures

• objective
  .name
  .syn or .neg
  .loop_count
  .value  # any structure you want

• ASA_locator
  .locator  # normally IPv6 address
  .protocol # IPPROTO_TCP or IPPROTO_UDP
  .port
  .etc
Simplified summary of calls (1)

- register_asa(asa_name)
- register_objective(objective)
- discover(objective)
Simplified summary of calls (2)

- `request_negotiate(objective, peer)`
- `listen_negotiate(objective)`
- `negotiate_step(objective)`
- `negotiate_wait(timeout)`
- `end_negotiate(result, reason)`
Simplified summary of calls (3)

- synchronize(objective, peer)
- listen_synchronize(objective)
- flood(objectives)
Recent changes

• Integrated various changes to GRASP
• Changed to integer error code returns
• Required implementations to accept objective values in CBOR
• Added non-blocking versions of each blocking call
Need help

• Mapping to Python was easy
• Need help on developing a robust mapping to C
  - Early draft of header file at
    https://www.cs.auckland.ac.nz/~brian/graspy/graspi.h
Discussion + next steps

• Comments? Questions?
• Should the WG take up this topic?