Topics

• Purpose of draft
• Overview
• Recent changes
• Main points
• Discussion, next steps
Purpose

• This document is intended to guide ASA writers in the general design of their code.
  - We expect ASAs to be written by a wide variety of programmers, specialised in the autonomic function concerned.
  - They are not expected to be GRASP experts. An API description will not be enough.
Overview of contents

- Logical Structure of an ASA
- Interaction with the Autonomic Infrastructure (ANI)
- Design of GRASP Objectives
- Life Cycle
- Coordination [TBD]
- Robustness
- Security Considerations
Recent changes

• Added details of event-loop scenario
• Clarified 'dry run' usage
• Filled in Life Cycle section
  – added co-authors
• Added Robustness section
Main points (1)

- ASAs run within the ACP and use GRASP
- ASAs should be multi-threaded but may use an 'event loop' structure:
  - must be self-monitoring & self-restarting
  - threads for flooding, synchronizing or negotiating each supported objective
  - thread to manage subsidiary non-autonomic devices
Main points (2)

- GRASP objectives follow GRASP rules
  - GRASP provides no transactional integrity. Locks and atomicity are the job of the ASA.
  - The 'value' of an Objective is only limited by CBOR; virtually any data structure is OK.
Main points (3)

- In a continuously running system, ASAs need systematic life cycle support.
Discussion + next steps

• Comments? Questions?
• Should the WG adopt this draft?