

A Reference Model for Autonomic Networking

draft-behringer-anima-reference-model-03.txt

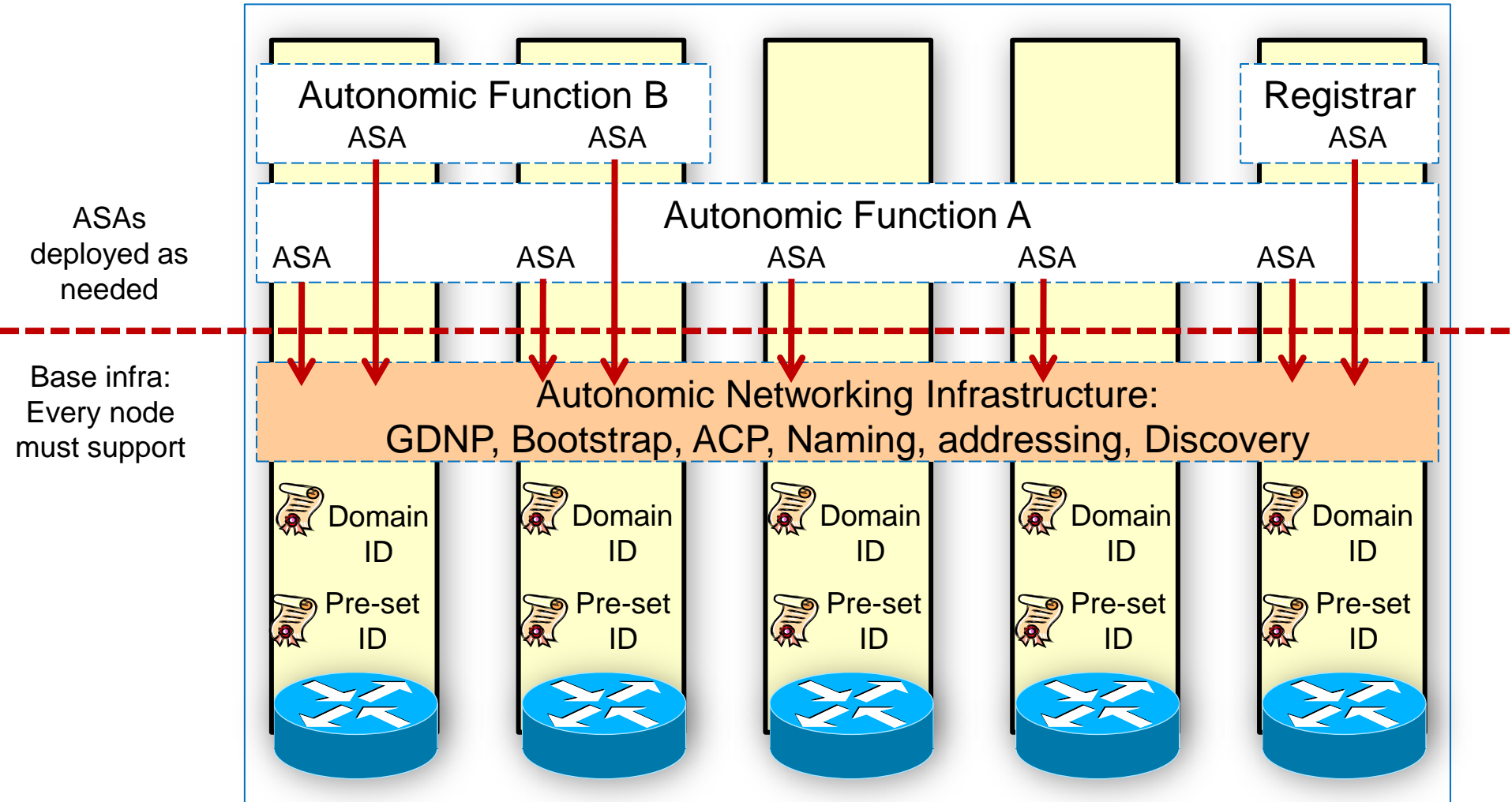
93rd IETF, 20 July 2015

Michael Behringer

Brian Carpenter

Toerless Eckert

Reference Model – High Level View



Network with autonomic functions

draft-behringer-anima-reference-model-03.txt

- 1. Introduction
 - 2. The Network View
 - 3. The Autonomic Network Element
 - 3.1. Architecture
 - 3.2. Full AN Nodes
 - 3.3. Constrained AN Nodes (*)
 - 4. The Autonomic Networking Infrastructure
 - 4.1. Naming
 - 4.1.1. Naming requirements
 - 4.1.2. Proposed Mechanisms
 - 4.2. Addressing
 - 4.2.1. Requirements and Fundamental Concepts
 - 4.2.2. The Base Addressing Scheme
 - 4.2.3. Possible Sub-Schemes
 - 4.2.4. Address Hierarchy
 - 4.3. Discovery
 - 4.4. Signaling Between Autonomic Nodes
 - 4.5. Intent Distribution
 - 4.6. Routing
 - 4.7. The Autonomic Control Plane
 - 5. Security and Trust Infrastructure
 - 5.1. Public Key Infrastructure
 - 5.2. Domain Certificate
 - 5.3. The MASA
 - 5.4. Sub-Domains (*)
 - 5.5. Cross-Domain Functionality (*)
-
- The diagram consists of several blue-bordered rectangular boxes on the right side of the page, each containing a list of changes or notes. Blue lines (callouts) point from these boxes to specific sections in the document outline on the left. For example, the top box points to sections 3.1, 3.2, and 3.3. The second box points to section 4.1. The third box points to section 4.2. The fourth box points to section 4.3. The fifth box points to section 4.7. The sixth box points to sections 5.1, 5.2, 5.3, 5.4, and 5.5.
- Moved MASA to “trust infrastructure”, and registrar to “ASA” section.
 - Introduced constrained node
 - Naming: New section, needs discussion and review
 - Addressing: Merged the addressing draft here, with some changes. Needs more discussion and review.
 - Discovery, signalling and intent distribution have new text, needs review.
 - Points to ACP draft. Should probably have more explanation here.
 - Ordered several “loose” bits into this section.

draft-behringer-anima-reference-model-03.txt

6.	Autonomic Service Agents (ASA)	16
6.1.	General Description of an ASA	
6.2.	Specific ASAs for the Enrolment Process	
6.2.1.	The Enrolment ASA	
6.2.2.	The Enrolment Proxy ASA	
6.2.3.	The Registrar ASA	
7.	Management and Programmability	19
7.1.	How an AN Network Is Managed	
7.2.	Intent (*)	
7.3.	Aggregated Reporting (*)	
7.4.	Feedback Loops to NOC(*)	
7.5.	Control Loops (*)	
7.5.1.	Types of Control (*)	
7.5.2.	Types of Control Loops (*)	20
7.5.3.	Management of an Autonomic Control Loop (*)	21
7.5.4.	Elements of an Autonomic Control Loop (*)	22
7.6.	APIs (*)	22
7.6.1.	Dynamic APIs (*)	22
7.6.2.	APIs and Semantics(*)	
7.6.3.	API Considerations (*)	
7.7.	Data Model (*)	
8.	Coordination Between Autonomic Functions (*)	24
8.1.	The Coordination Problem (*)	
8.2.	A Coordination Functional Block (*)	
9.	Security Considerations	
9.1.	Threat Analysis	

- Clearly separate ASA from infrastructure now.
- New section on ASAs
- The registrar is now covered here, since it is an ASA

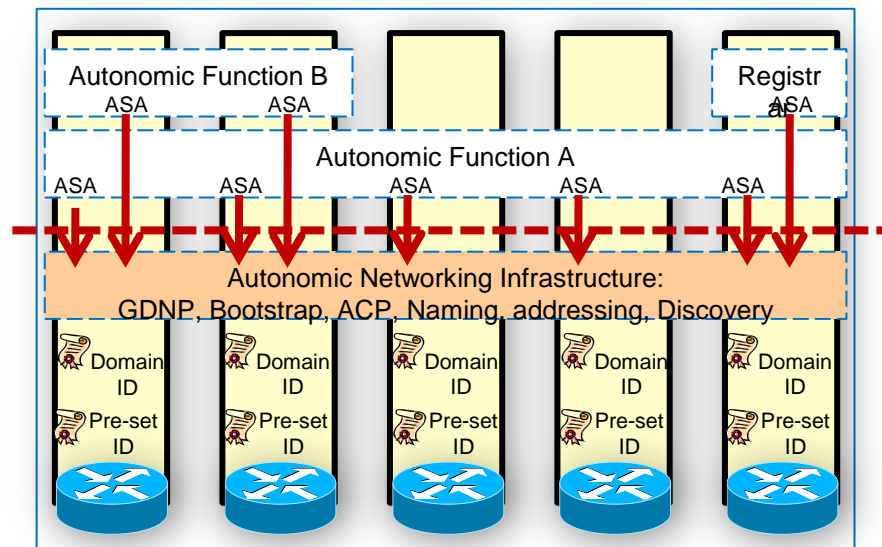
- New section, collecting some previously loose bits, and some new content. Needs reviews – how much detail do we want to put in here?

- New section about interactions of autonomic functions. More long term, but highly relevant.

- Needs more work.

Document Structure

- Structure of the document becoming stable
- No major issues with the structure itself



Network with autonomic functions

Naming

- **Why names?**
 - As an identity
 - As a subject name in the autonomic certificate
- **Structured names:**
 - Ex: Location-DeviceType-FunctionalRole-DistinguisherNumber@NameofDomain
 - Use self-knowledge for part of the name (e.g., device type)
 - Use other mechanisms (intent) for other parts (e.g., domain)
- **Open questions:**
 - Should we support assigned names, automatically created names, or both?
 - If automatic, how do we assign the names?

Addressing – Where to Cover?

- Used to be a separate draft ([draft-behringer-autonomic-addressing](#))
- But, this draft is not a standalone chartered item
- Request from WG chair was to integrate with an existing document
- Currently put the entire addressing doc into the reference draft.
 - Is this the right place? (for addressing schemes?)
- Possible way forward:
 - Leave requirements and concepts in reference draft
 - Put the addressing schemes into ... ? ACP draft?

Addressing - Scope

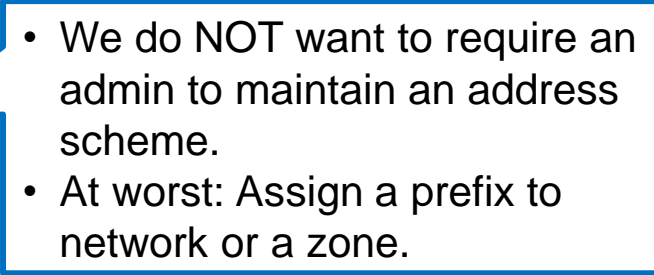
- **In scope: Addressing used by the Autonomic Networking Infrastructure (and indirectly by Autonomic Service Agents) inside an autonomic domain.**
- **Not in scope: Addressing of the data plane, i.e. anything that is used for services to customers.**
- **An autonomic function could negotiate address space for the data plane, for example draft-jiang-auto-addr-management.**
 - The function *uses* autonomic address space
 - But it *assigns and manages* data plane address space

• Is that sufficiently clear?

Addressing – Various points

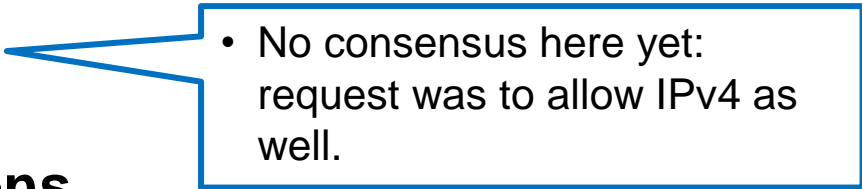
- **An Autonomic Node gets an address.**
 - **ASAs do NOT get addresses.**
 - **Autonomic nodes multiplex ASAs.**
- **Non-autonomic nodes do not get autonomic address**

Addressing - Requirements

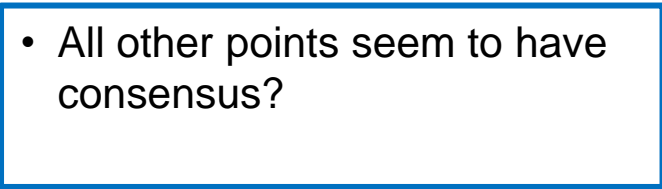
- **Zero-touch for simple networks**
 - **Low-touch for complex networks**
 - **Flexibility (allow for growth, splits, merges, etc)**
 - **Robustness (admin can't mess up)**
 - **Support for virtualization**
 - **Simplicity**
 - **Scale**
 - **Upgradability**
- 
- We do NOT want to require an admin to maintain an address scheme.
 - At worst: Assign a prefix to network or a zone.

Addressing - Concepts

- **IPv6 only (for the autonomic mechanisms)**
- **Usage: For autonomic functions exclusively**
- **Separation (from user address space)**
- **Overlay network**
- **Use ULA, on virtual interfaces**
- **No link addressing, only link local**
- **No external connectivity**



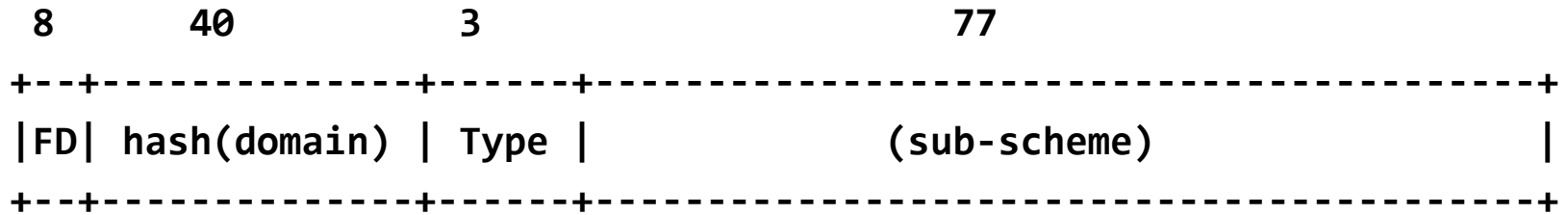
• No consensus here yet: request was to allow IPv4 as well.



• All other points seem to have consensus?

Addressing – Base Scheme

- **Base Scheme:**



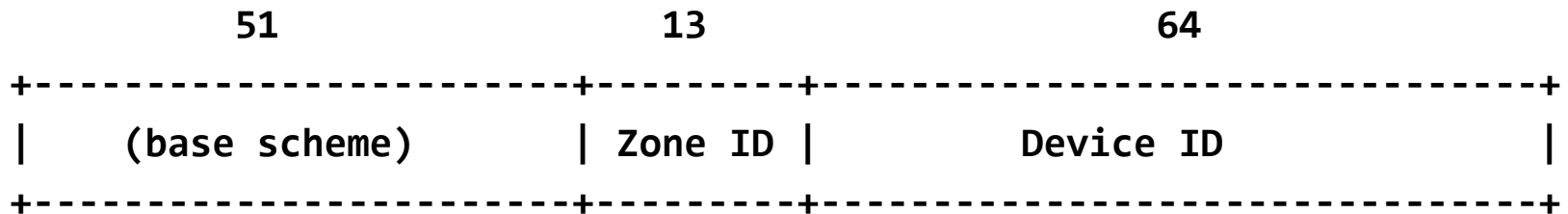
- **Hash(domain) provides pseudo-random prefix, as required by RFC4193 (ULA)**
- **We suggest a type field, to allow different address schemes in the future.**
- **Idea: Standardize only one type initially.**

- Do we agree so far?
- Comments? Concerns?

Addressing – Sub-Scheme 1

- Needs discussion

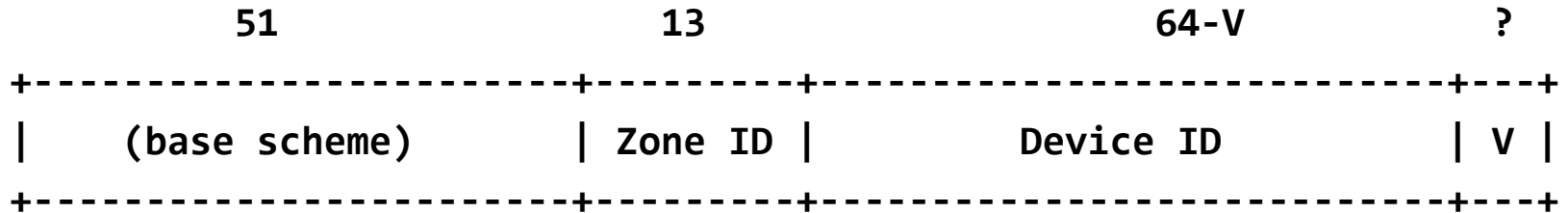
- **Sub-Scheme 1:**



- **Registrar assigns device ID**
 - It is unique for a device in a domain
 - It does NOT specify a locator, but an identifier
 - Device ID does not change in the lifetime of a device
- **Zone-ID initially zero.**
 - When aggregation is required, use a zone-ID $\langle \rangle 0$

Addressing – Sub-Scheme 2

- **Sub-Scheme 2:**



- **Add “Virtualisation” bits at the end**
 - Allow addressing various virtual machines on a single node
- **Keep routing simpler:**
 - Node announces not a /128, but for example /127

• Needs discussion

Discovery, Signaling, Intent Distribution

- **Overall goal: Minimise northbound interfaces. Thus:**
- **Discovery needed for discovering:**
 - Nodes
 - Services
- **Signalling needed to negotiate between nodes, synchronise, etc.**
- **Intent distribution should also be horizontal**
- **All these could be covered with a single protocol**
 - Candidate protocol: draft-carpenter-anima-gdn-protocol

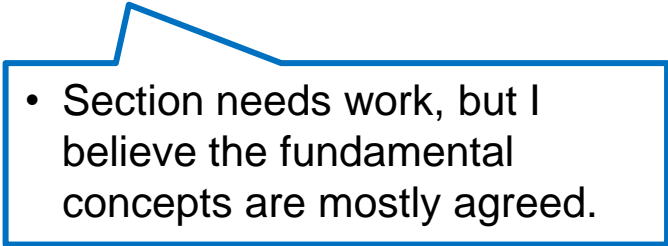
• Needs discussion

Routing and ACP

- **(covered in the ACP discussion)**

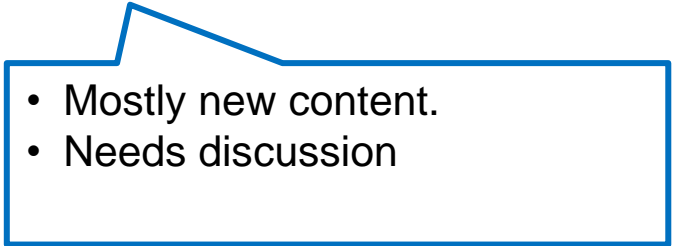
Security and Trust Infrastructure

- **Premise: “Self-protection”**
 - **Autonomic functions do not require configuration to be secure. They are designed and negotiated securely.**
- **Use a domain based PKI**
 - **Domain has a CA**
 - **The registrar(s) are RAs**
 - **MASA server allows for vendor certification.**

- 
- Section needs work, but I believe the fundamental concepts are mostly agreed.

Management Section

- **Generally “self-management”.**
- **Admin guidance through Intent**
- **Coexistence with existing methods (NETCONF, SNMP, SSH, etc)**
- **Conflict resolution**
- **Aggregated reporting**
- **Feedback loops to NOC**
- **Control loops**
- **APIs**

- 
- Mostly new content.
 - Needs discussion

Coordination between Autonomic Functions

- **Autonomic functions may interact:**
 - Dependency, conflict, cooperation
- **Various ways to deal with interactions, at various times:**
 - Build time
 - Deploy time
 - Run time
- **Require a coordination function**

Summary

- **Structure getting solid**
- **Content still needs work**
- **Open, high level question: How much detail, how much forward looking items (to be seen over time)**
- **Adoption as WG document?**