JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

11100 Johns Hopkins Road
Laurel, MD 20723-6099

DTN Network Management

Edward J. Birrane, Ph.D.
Johns Hopkins University, Applied Physics Laboratory (JHU/APL)
Edward.Birrane@jhuapl.edu
(W) 443-778-7423

History

- Examined uniqueness of the problem, 2011-2013
  - Some early pubs defining the problem as related to DTN
    - *Birrane, E, & Cole, R. (2011). Management of Disruption-Tolerant Networks: A Systems Engineering Approach.*
    - *E. Birrane, S. Burleigh, V. Cerf, "Defining Tolerance: Impacts of Delay and Disruption when Managing Challenged Networks,"*
    - *E. Birrane, H. Kruse, "Delay-Tolerant Network Management: The Definition and Exchange of Infrastructure Information in High Delay Environments"*
  - Reviewed popular engineering approaches
    - *Autonomous fault protection schemes*
    - *Mobile code and scripting schemes*
    - *Spacecraft telemetry schedules*
    - *Deterministic rule-based expert systems*

- Delay-Tolerant Network Management Protocol (DTNMP) 2013
  - Published to DTNRG, Initial implementation by NASA
  - Utility outside of NASA network management

- Renamed as Asynchronous Management Protocol (AMP) 2015
  - Submitted as set of IDs to DTNWG.

How Do We Manage Networks Today? Do we need a new thing?

Low-latency approaches to network management fail to scale with increasing delays and disruptions.

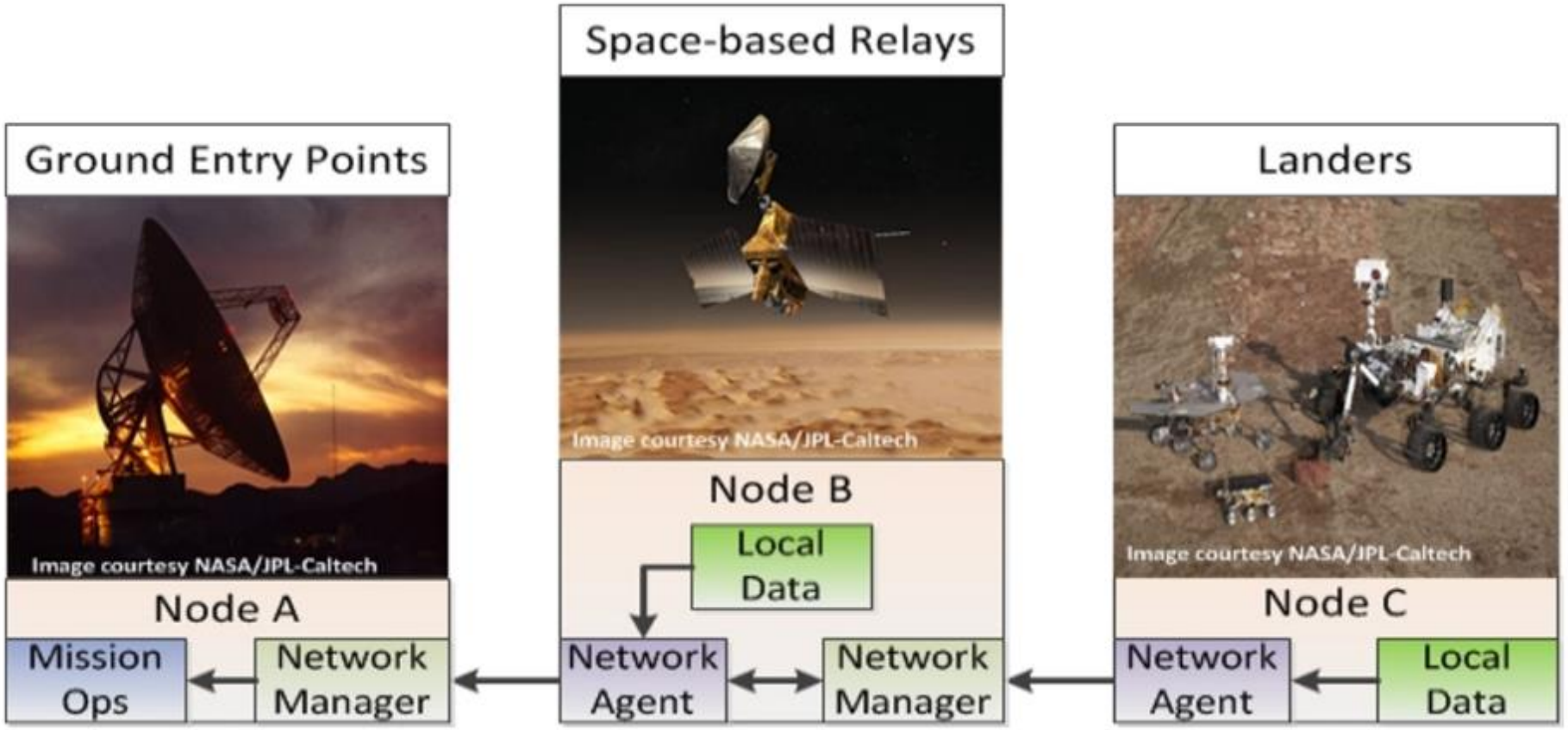
- Rich set of evolving capabilities
  - Simple Network Management Protocol (SNMP).
    - *Pull model of information from managed devices.*
      - Support for “traps” to push unreliable notifications of pre-defined events.
  - Network Configuration Protocol (NETCONF).
    - *XML-based, session-based remote-procedure call (RPC) interface for node configuration.*
  - Remote Network Monitoring MIB (RMON).
    - *Mechanisms for exchanging network monitoring data.*
- Poor scaling with delays, disruptions, or commanding
  - Focus on getting data to operators.
  - Less focus on in-situ response options.
  - Reliance on scripting and mobile code which is not always a deployment option.

New capabilities have come online such as RESTConf, Yang 1.1, and YangPush.

These capabilities add more “push” semantics, but still struggle to operate in highly disrupted scenarios. The AMA is being reworked to focus more effectively on the unique characteristics of challenged networks.

Asynchronous Management Architecture

Let's think through an extreme use case…



Intersection: Spacecraft and Data Center?

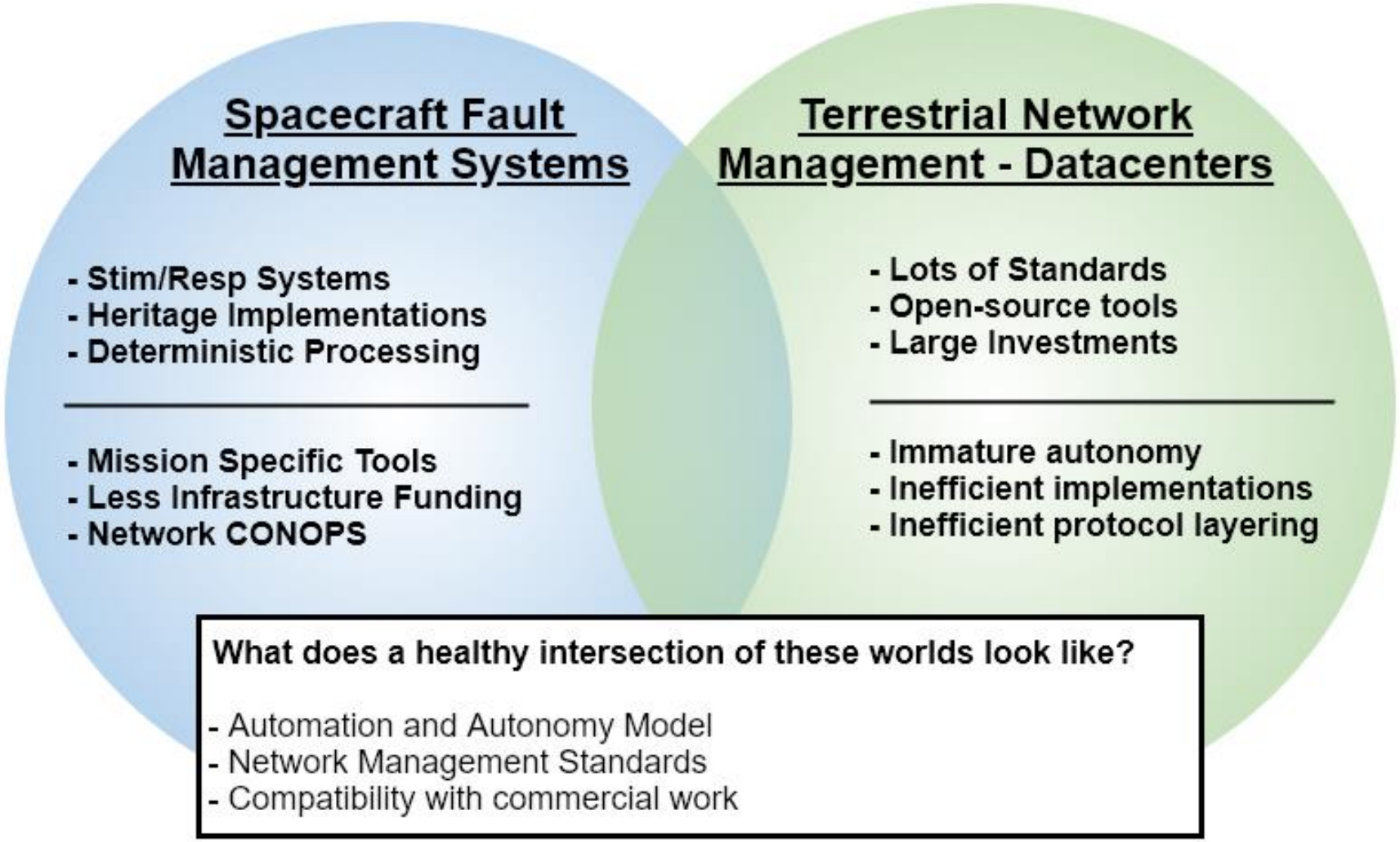
How do we manage partitioned networks, such as:

- New Space Networks
- Challenged Sensor Networks
- Vehicle Networks
- Resource-Constrained Networks
- Disruption-Tolerant Networks

Spacecraft have this problem, what do they do?

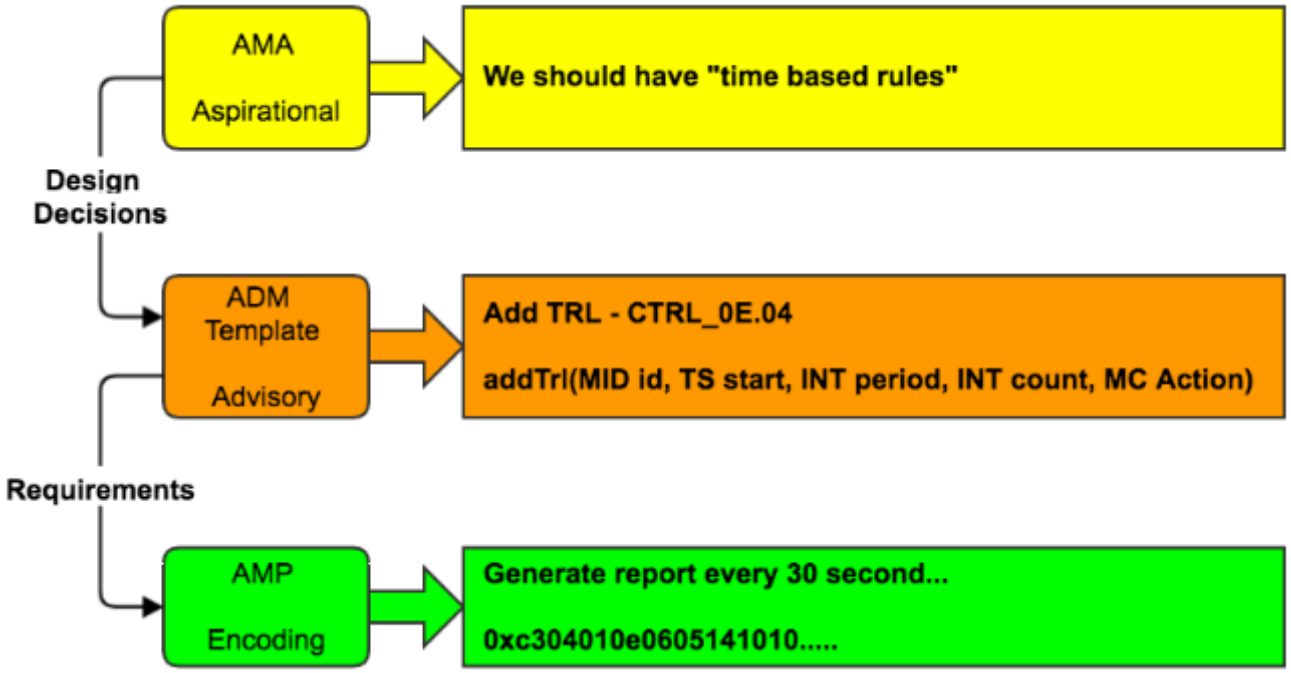
- Autonomy models
- VERY efficient encodings
- Low requirements for processing.

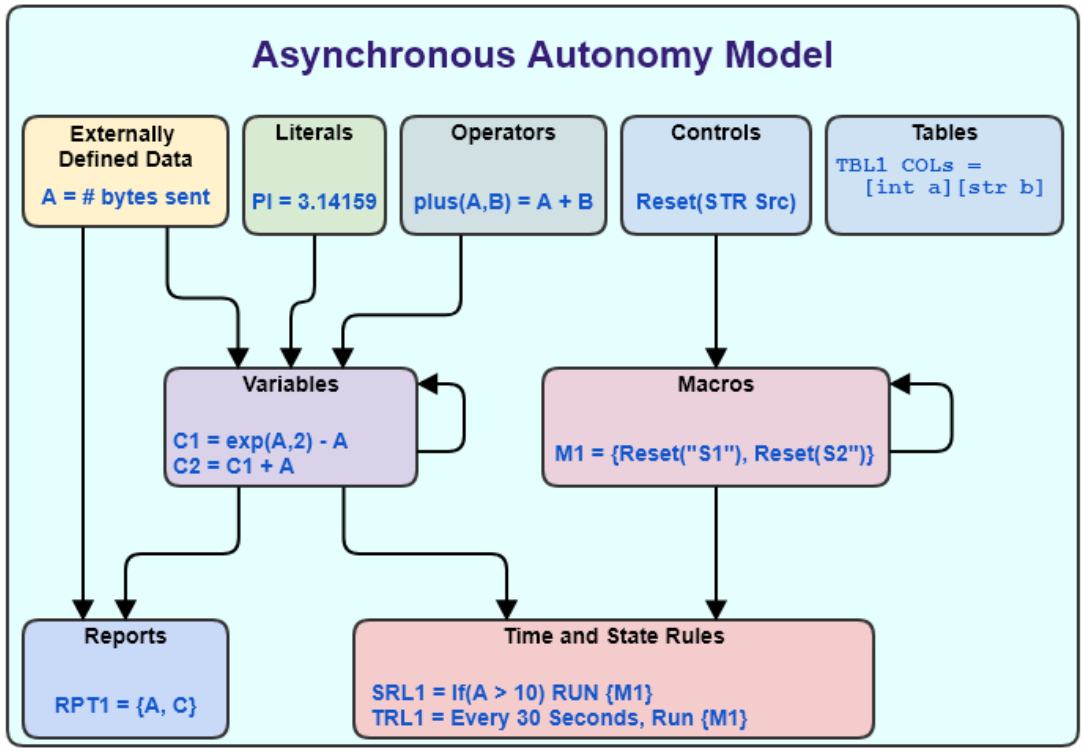
Can we bring these worlds together in an IETF-useful way?



AMA/ADM/AMP Interactions

Asynchronous Management is network management via configured, deterministic autonomy.





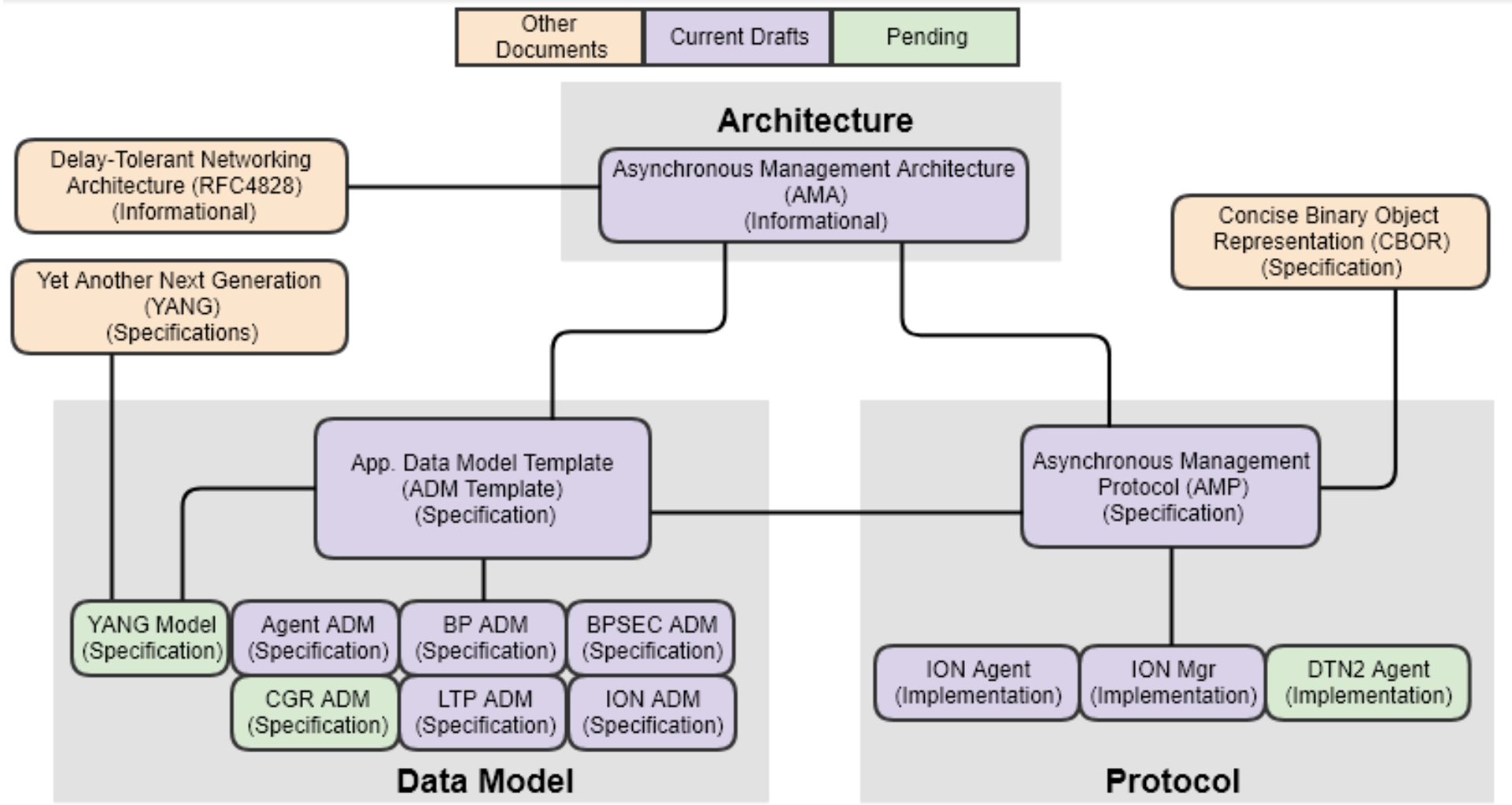
It is defined as:

(1) An Architecture - https://datatracker.ietf.org/doc/html/draft-ietf-dtn-ama-03

(2) A Data Model - https://tools.ietf.org/html/draft-birrane-dtn-adm-03

(3) A Protocol - https://datatracker.ietf.org/doc/html/draft-birrane-dtn-amp-08

Work is ongoing in the DTN-WG



Active Drafts (For Reference)

- https://datatracker.ietf.org/doc/draft-birrane-dtn-ama/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-amp/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-agent/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-bp/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-bpsec/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-ion-bpadmin/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-ion-ipnadmin/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-ion-ltpadmin/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-ionadmin/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-ionsec/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-adm-ltp/
- https://datatracker.ietf.org/doc/draft-birrane-dtn-ampmgr-sql/

AMA: Overview

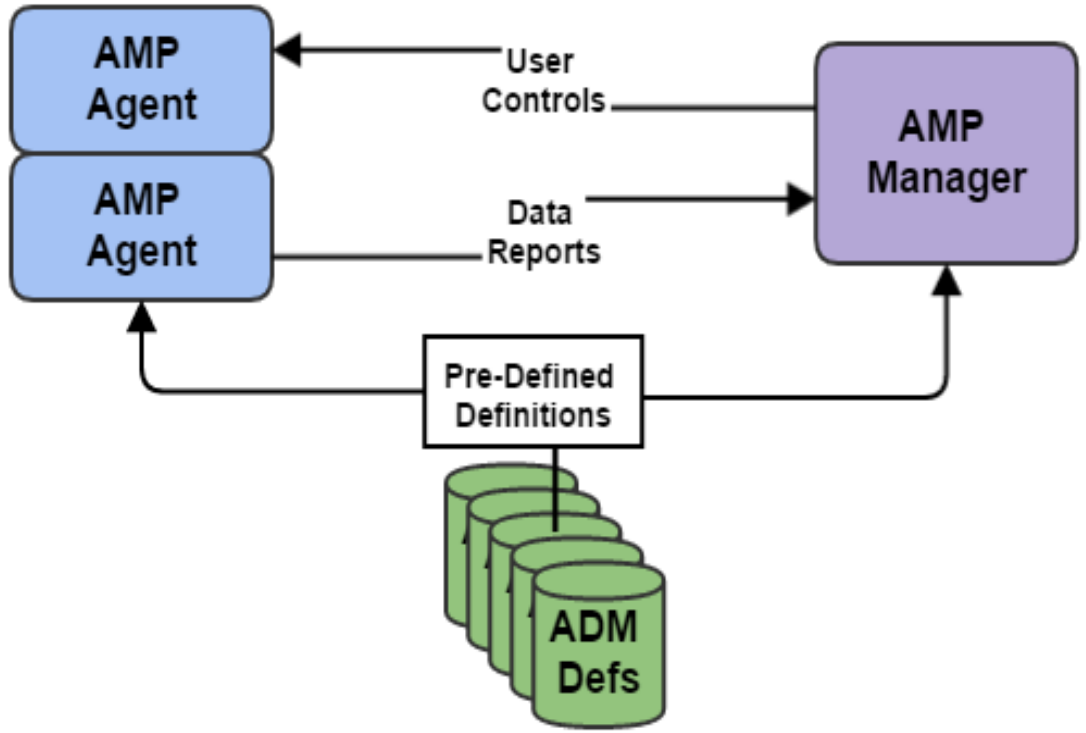
From https://datatracker.ietf.org/doc/html/draft-ietf-dtn-ama-03

- Service Definitions
  - **Configuration**: Change settings on an Agent.
  - **Reporting**: Receive performance information from an Agent.
  - **Autonomous Parameterized Control**: Change Agent Behavior.
  - **Administration**: Fine-grained access to abilities.

- Desirable Properties
  - **Intelligent Information Push**: Can’t rely on others.
  - **Minimize Message Size**: Increase probability of delivery.
  - **Absolute Data Identification**: pre-shared, global naming when possible.
  - **Custom Data Definition**: Only send minimal necessary data sets.
  - **Autonomous Operation**: Decisions local to Agent based on its config.

AMA: The Simple System Model

- Agents
  - Run on Managed Devices
  - Configure/Report on devices
  - Heavy autonomy and parameterized control
- Manager(s)
  - Collect/Fuse data from Agents
  - Configure Agent behavior
  - Open-loop control
- ADMs
  - Well-named Data and Controls
  - Superset of MIB
  - Move to describe them in YANG
  - Preconfiguration reduces msg size



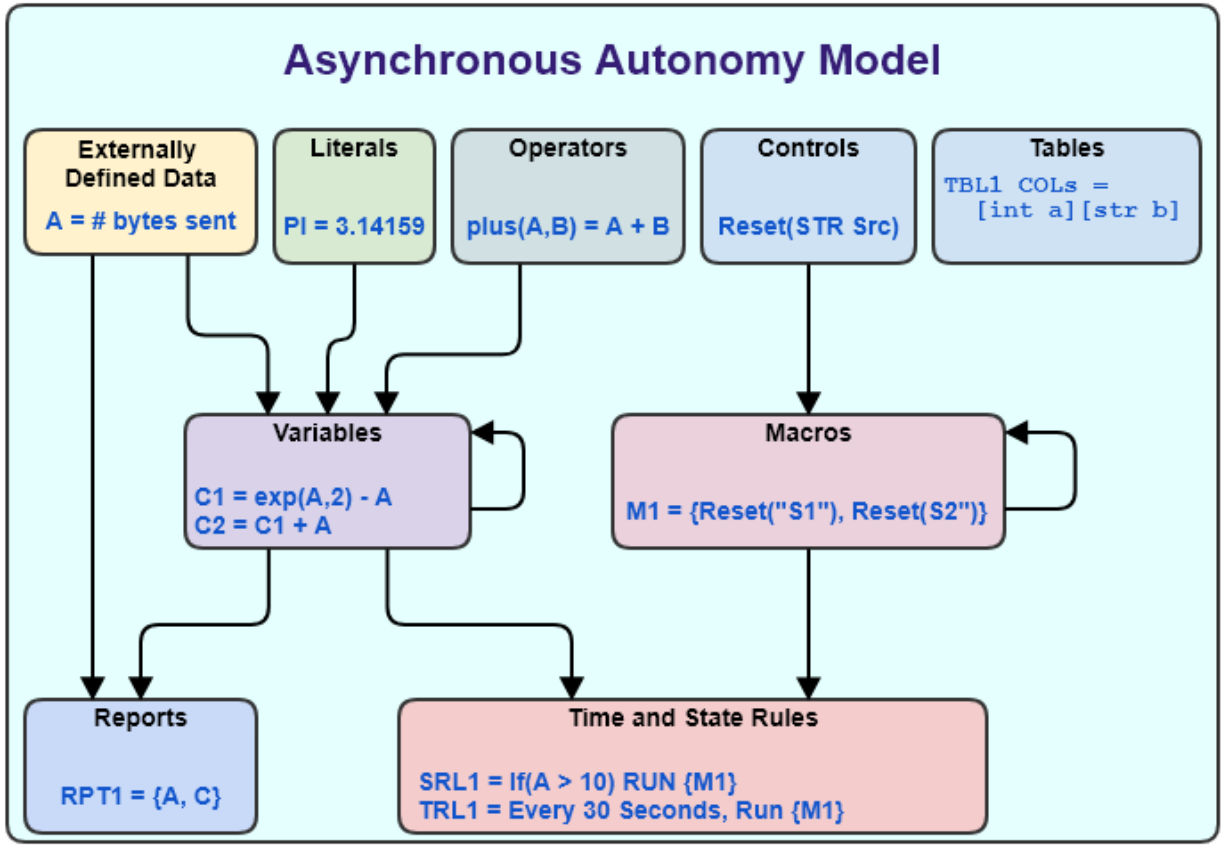
ADM Template: Logical Modeling

- Separate the data specification from its encoding.
  - Use AMP specification to define how to compactly encode ADM items
- ADMs Schemas will define logical models
  - Designed to identify minimum set of information per data model
  - Remove any “encoding hints” from the models.
  - Use the YANG modelling language
    - *Tools exist to validate YANG schemas for correctness and plot dependencies.*
- ADMs will be defined in JSON
  - Conventions will be defined to make JSON writing expressive and “easy”
  - Reuse existing notations/delimiters where possible (query string)
- Define compilers/adapters
  - Presuppose adapters/compilers to generate encodings as necessary

Asynchronous Management Model (AMM)

- "Atomic" Elements
  - **EDDs**: collected by agents.
  - **Literals**: useful constants.
  - **Ops**: opcodes for math functions.
  - **Ctrls**: opcodes for agent behavior.
  - **Tables**: Structured data sets
- "Dynamic" Elements
  - **Vars**: strong-typed variables
  - **Macro**: Ordered set of Ctrls.
  - **Rpts**: Ordered sets of data
  - **Rules**: Time or State based autonomy.

An AMM defines 9 types of data for each application/protocol managed in the AMA.



BP Example: Externally Defined Data

```
”edd": [
      {
        "name": "num_pend_reassembly",
        "type": "UINT",
        "description": "number of bundles pending reassembly"
      },

      {
        "name": "bundles_by_priority",
        "type": "UINT",
        "parmspec": [{"type":"UINT", "name":"mask"}],
        "description": ”Number of bundles for the given priority. Priority is given as a priority mask where
                        Bulk=0x1, normal=0x2, express=0x4. Any bundles matching any of the masked
                        priorities will be included in the returned count"
      },
```

BPSec Example: Tables

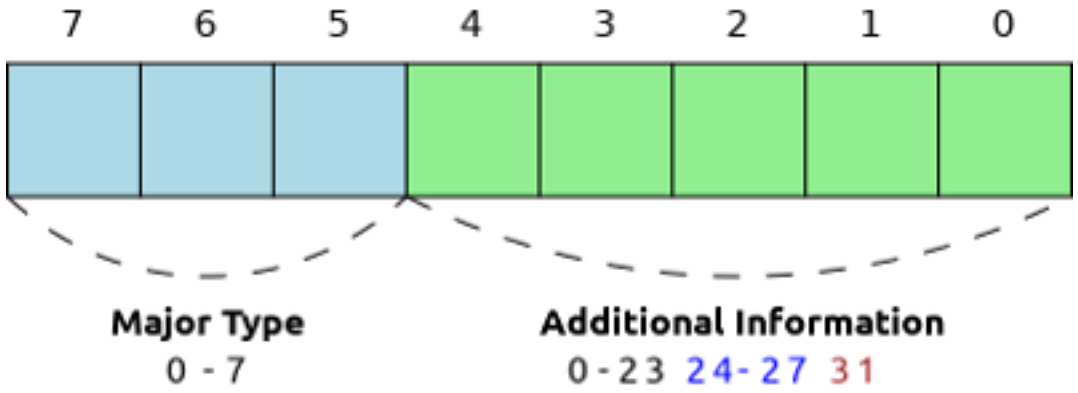
```
"table-templates" : [
    {
      "name": "ciphersuites",
      "columns": [{"type":"STR",  "name":"csname"}],
      "description": "This table lists supported ciphersuites."
    },

    {
      "name": "bib_rules",
      "columns": [{"type":"STR",        "name":"SrcEid"},
                  {"type":"STR",        "name":"DestEid"},
                  {"type":"UINT",       "name":"TgtBlk"},
                  {"type":"STR",        "name":"csName"},
                  {"type":"STR",        "name":"keyName"}
              ],
      "description": "BIB Rules."
    },
```

IonAdmin Example: Controls

```
“controls”: [
{
    "name": "node_contact_add",
    "parmspec": [
      {"type":"TS",          "name":"start"},
      {"type":"TS",          "name":"stop"},
      {"type":"UINT",        "name":"node_id"},
      {"type":"STR",         "name":"dest"},
      {"type":"FLOAT32",     "name":"data_rate"},
      {"type":"FLOAT32",     "name":"prob"}
    ],
    "description": "This control schedules a period of data transmission from node_id to dest. The period of transmission will begin at start and end at stop, and the rate of data transmission will be data_rate bytes/second. Our confidence in the contact defaults to 1.0, indicating that the contact is scheduled - not that non-occurrence of the contact is impossible, just that occurrence of the contact is planned and scheduled rather than merely imputed from past node behavior. In the latter case, confidence indicates our estimation of the likelihood of this potential contact."
  },
```

Asynchronous Management Protocol

A CBOR encoding of ADM information.



Format	[1, [2, 3]]	[_ 1, [2, 3]]
RFC 713	c2 05 81 c2 02 82 83	
ASN.1 BER	30 0b 02 01 01 30 06 02 01 02 02 01 03	30 80 02 01 01 30 06 02 01 02 02 01 03 00 00
MessagePack	92 01 92 02 03	
BSON	22 00 00 00 10 30 00 01 00 00 00 04 31 00 13 00 00 00 10 30 00 02 00 00 00 10 31 00 03 00 00 00 00 00	
UBJSON	61 02 42 01 61 02 42 02 42 03	61 ff 42 01 61 02 42 02 42 03 45
CBOR	82 01 82 02 03	9f 01 82 02 03 ff

AMP is:

- Independent of BP
  - AMP messages can use any transport.

This is a content encoding, and not necessarily a transport encoding.

- A binary encoding of ADM Information
  - CBOR is used because it is compact and well supported.
  - CBOR encoders can be < 10KB in size.

- A set of messages to exchange these encodings
  - Simple messaging, like SNMP and NETCONF
    - *Register Agent*
    - *Report Set*
    - *Perform Control*
  - Most AMP Agent control is captured in an AMP Agent ADM.

- A state machine
  - Describing the behavior of agents and managers in a network.

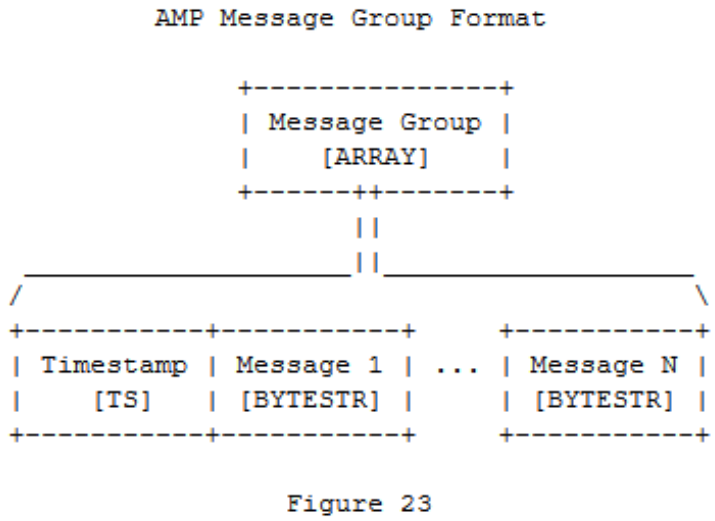
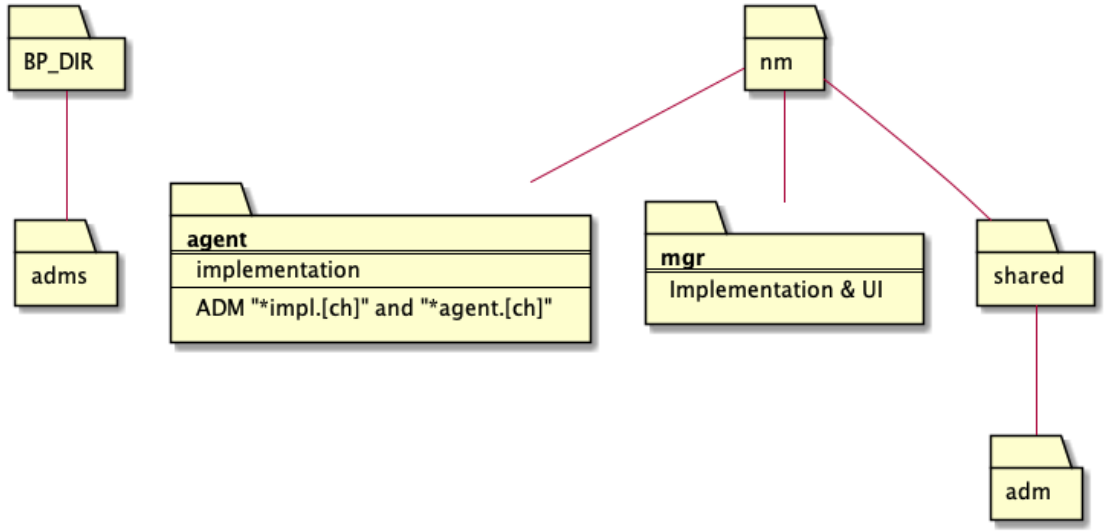


Figure 23

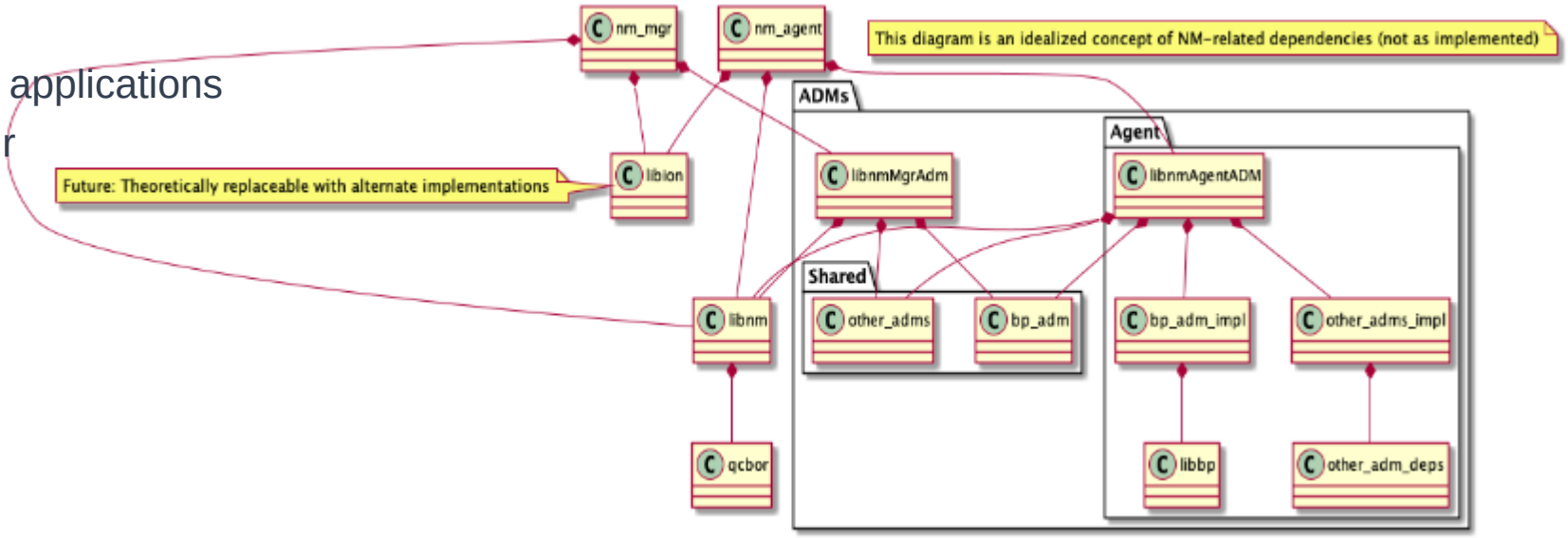
ION Reference Implementation

AMP Agent, AMP Text-Based Manager

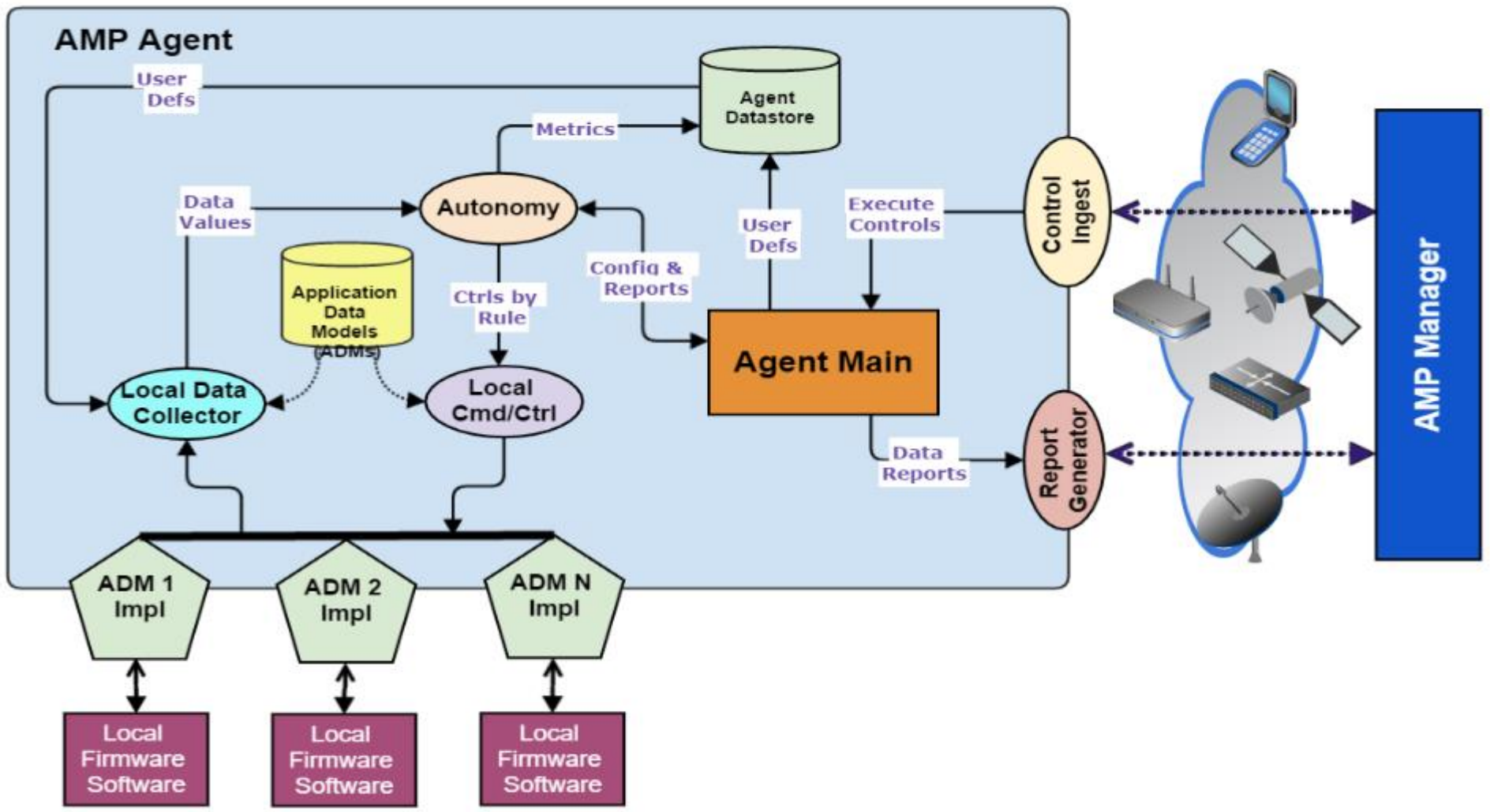
- AMP in ION Release 4.0
  - Updated directory structure
  - Support for BPv6 and BPv7
  - Feature complete to latest AMP spec
    - https://tools.ietf.org/html/draft-birrane-dtn-amp-08



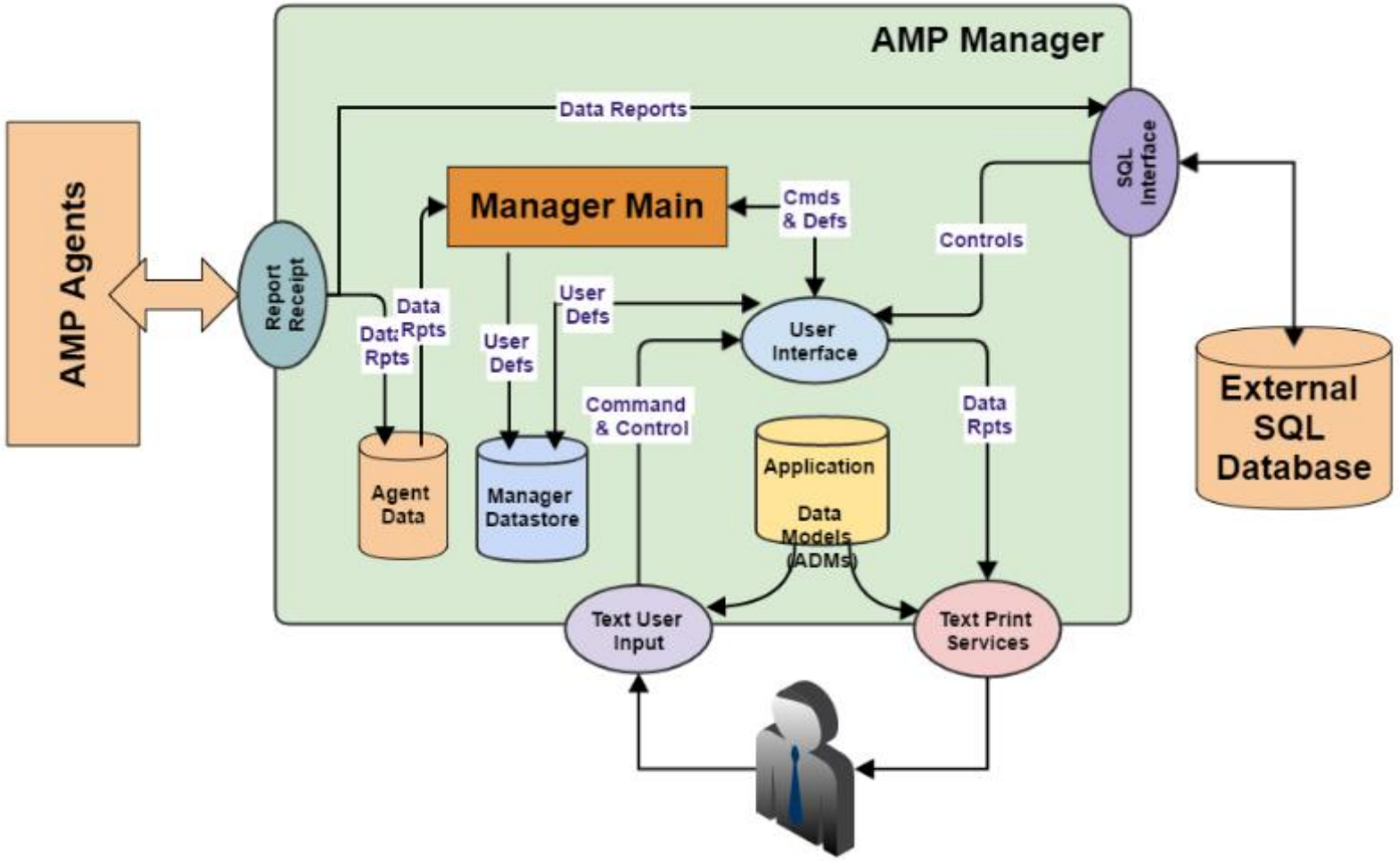
- What's New?
  - New directory design
    - ADMs can be stored with their applications
    - Linked as needed to agent/mgr
  - Endian-Neutral CBOR
    - qcbor
  - Test Support
    - Additional regression tests



AMP-ION Agent

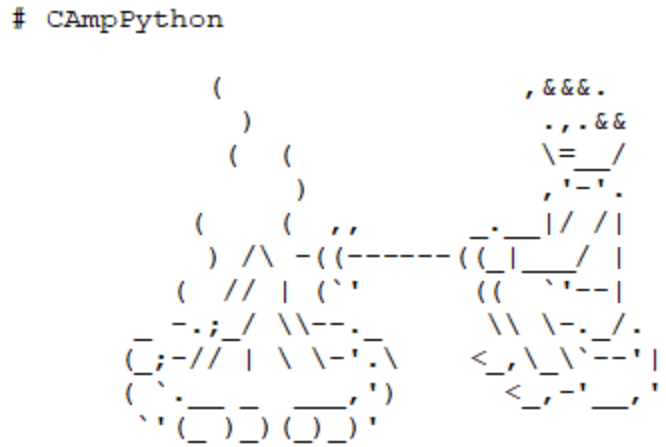


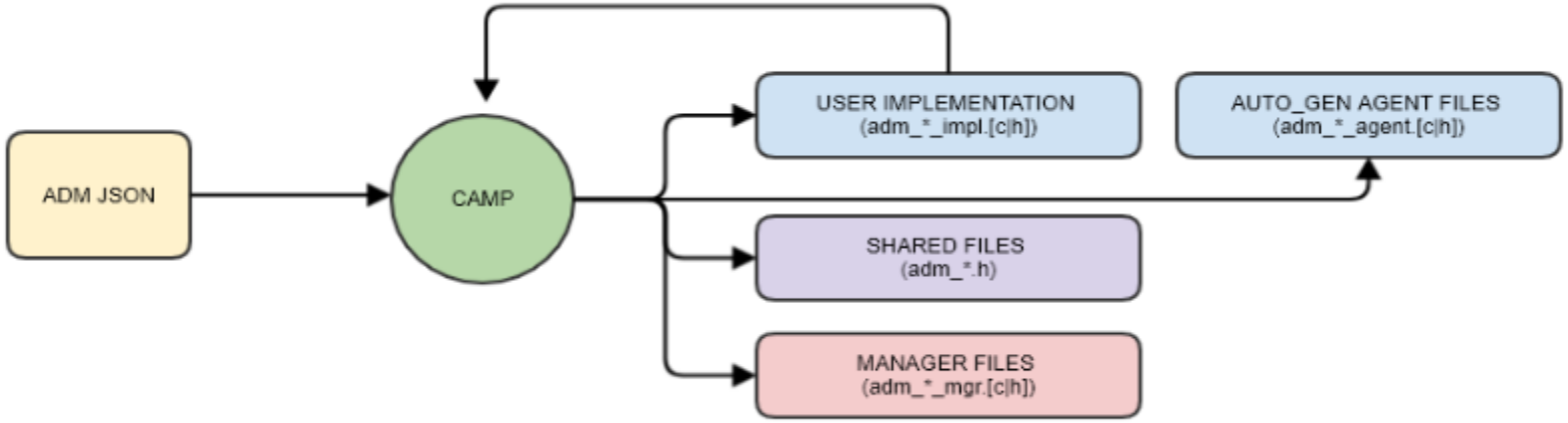
AMP-ION Manager



ADM Auto Generation

- ION C-generating AMP Python Script (CampPython)
  - Github project
    - *C API for AMP defined for ION (3.x or 4.x line)*
    - *Produces .c/.h files per ADM*
    - *Includes a user-customizable .c/.h) and "round-tripping"*
    - *camp <adm.json> -c <old_impl.c> -s <old_impl.h>*
  - Adding and maintaining ADMs much simplified
    - *But, now, lots of data. 200+ data items, ~100 controls/operators*



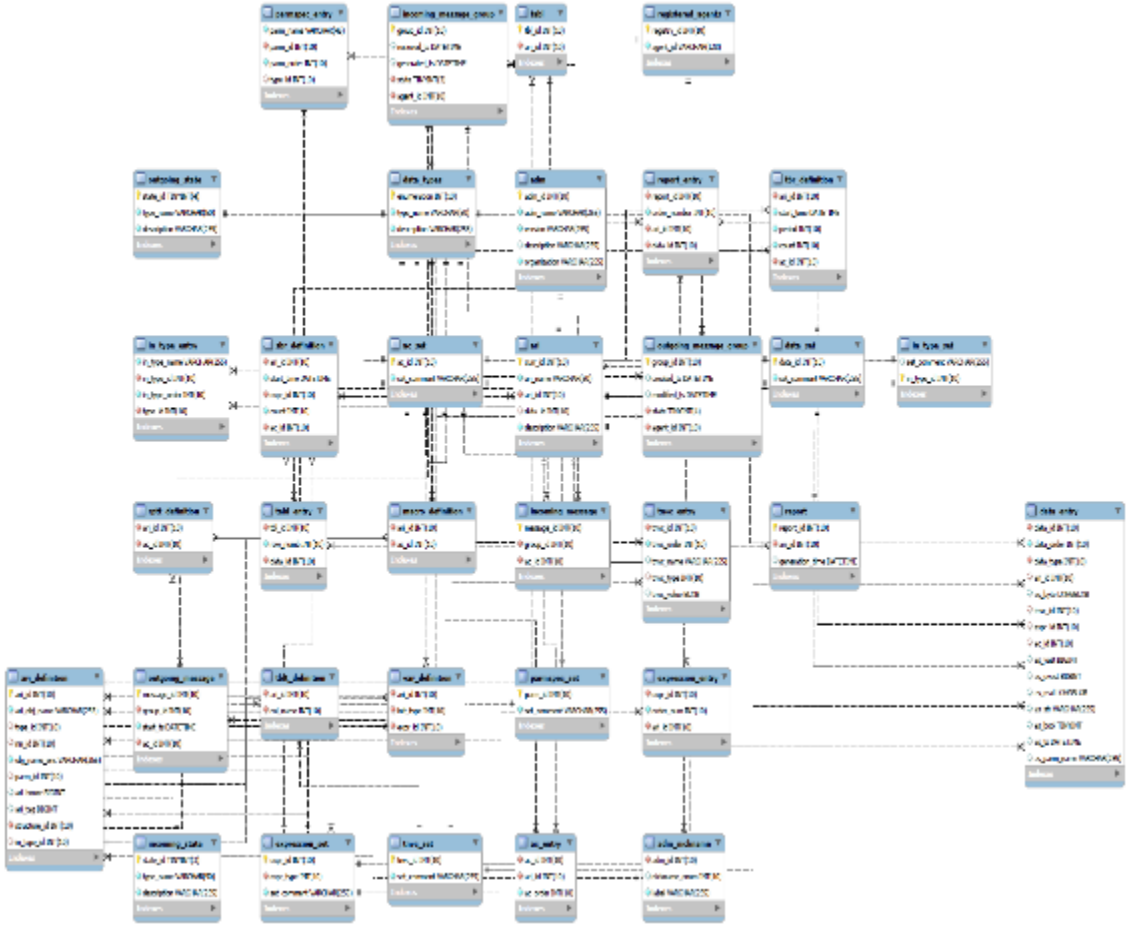


CAMP User Implementation File Example

```
value_t adm_bpsec_get_ciphersuite_names(tdc_t params)
{
        value_t result;
   /* +--------------------------------------------------------------------+
    * |START CUSTOM FUNCTION get_ciphersuite_names BODY
    * +--------------------------------------------------------------------+/

        char *tmp = bpsec_instr_get_csnames();
        result.value.as_ptr = STAKE(size));
        memcpy(result.value.as_ptr, tmp, size);
        SRELEASE(tmp);
        result.type = AMP_TYPE_STRING;

   /* +--------------------------------------------------------------------+
    * |STOP CUSTOM FUNCTION get_ciphersuite_names BODY
    * +--------------------------------------------------------------------+/
        return result;
}
```

AMP Manager SQL Schema

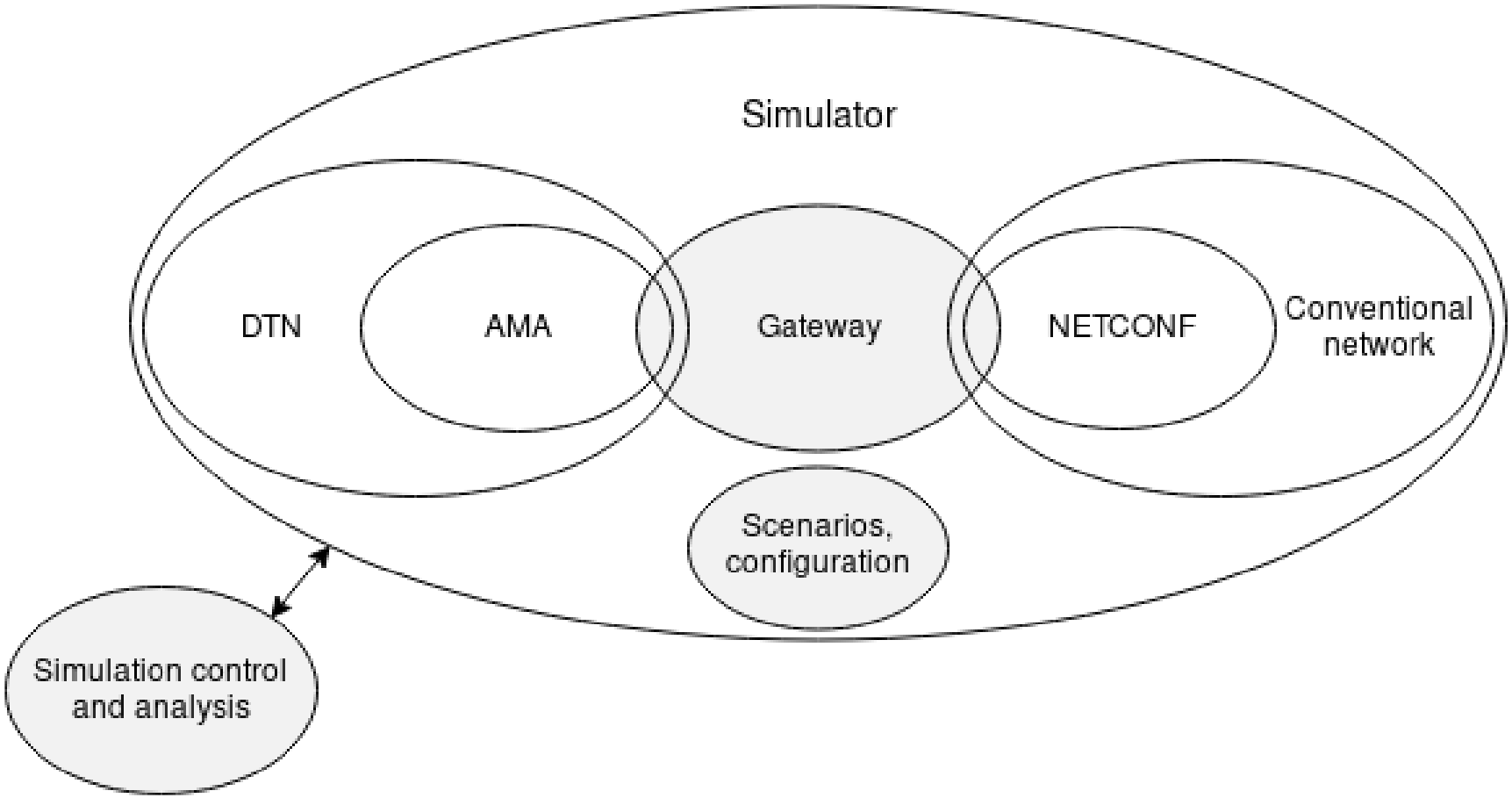
- AMP Schema
  - Abstracts any binary encoding from user.
  - SQL-enabled AMP Manager:
    - *Polls DB for new info to send to AMP Agents.*
    - *Stores received agent information for visualization.*
- AMP-Specific Tables
  - Data representations for constants, literals
  - Static ADM Information, Operator-defined data
  - Outgoing/Incoming message groups
- CAMP integration
  - CAMP generates SQL files from JSON ADMs
- Testing
  - SQL scripts to auto-populate tables for testing



Master's Thesis: NETCONF <-> AMP

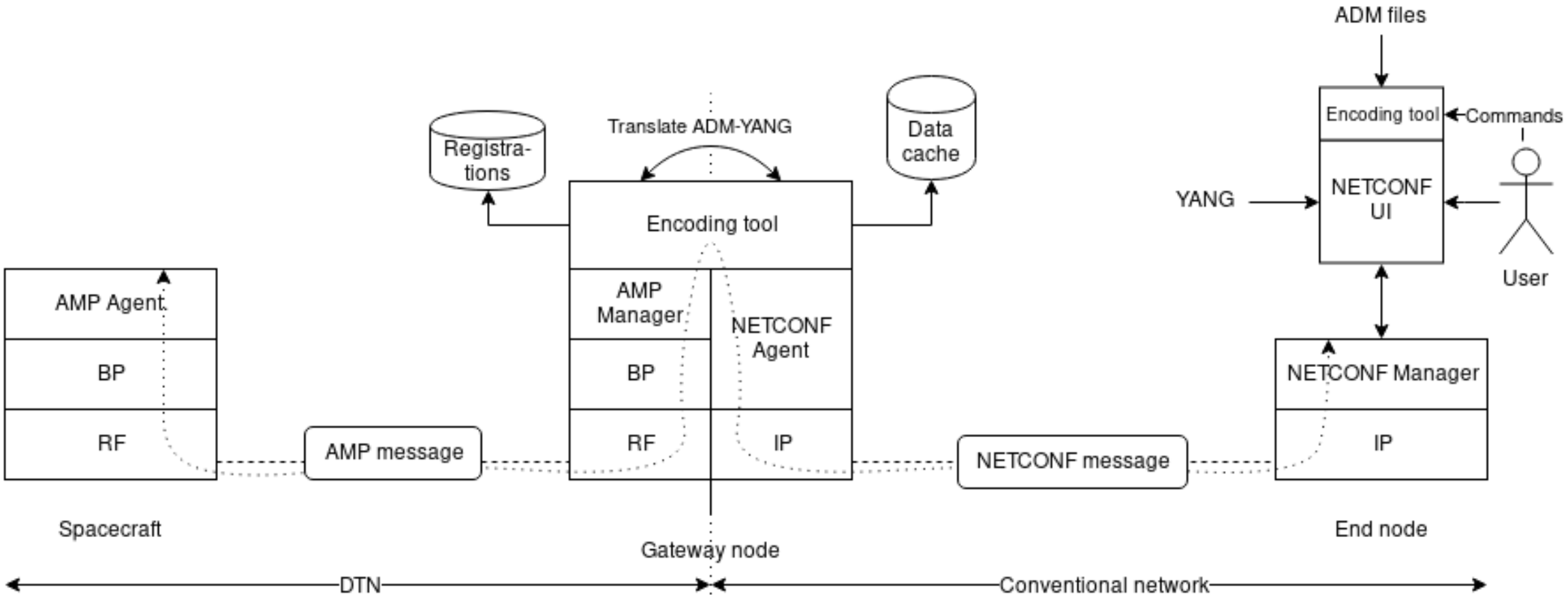
- Jean Chorin
  - Visiting UMBC from Dresden (works for Marius)
- NETCONF is "the" current NM protocol
  - YANG is its modelling language
- NETCONF isn't a DTN solution
  - Requires sessions and state. No Autonomy. XML encoding. Can't send delta updates. Chatty, closed-loop control CONOPS.
  - No S/C is going to run a NETCONF agent.
- ADM <-> YANG converter
  - Model mapping. Scripts to populate NETCONF stores from DTN.
  - Handles asynchronous-to-synchronous data issues.

External Work: Integrate with NETCONF



From: Jean Chorin

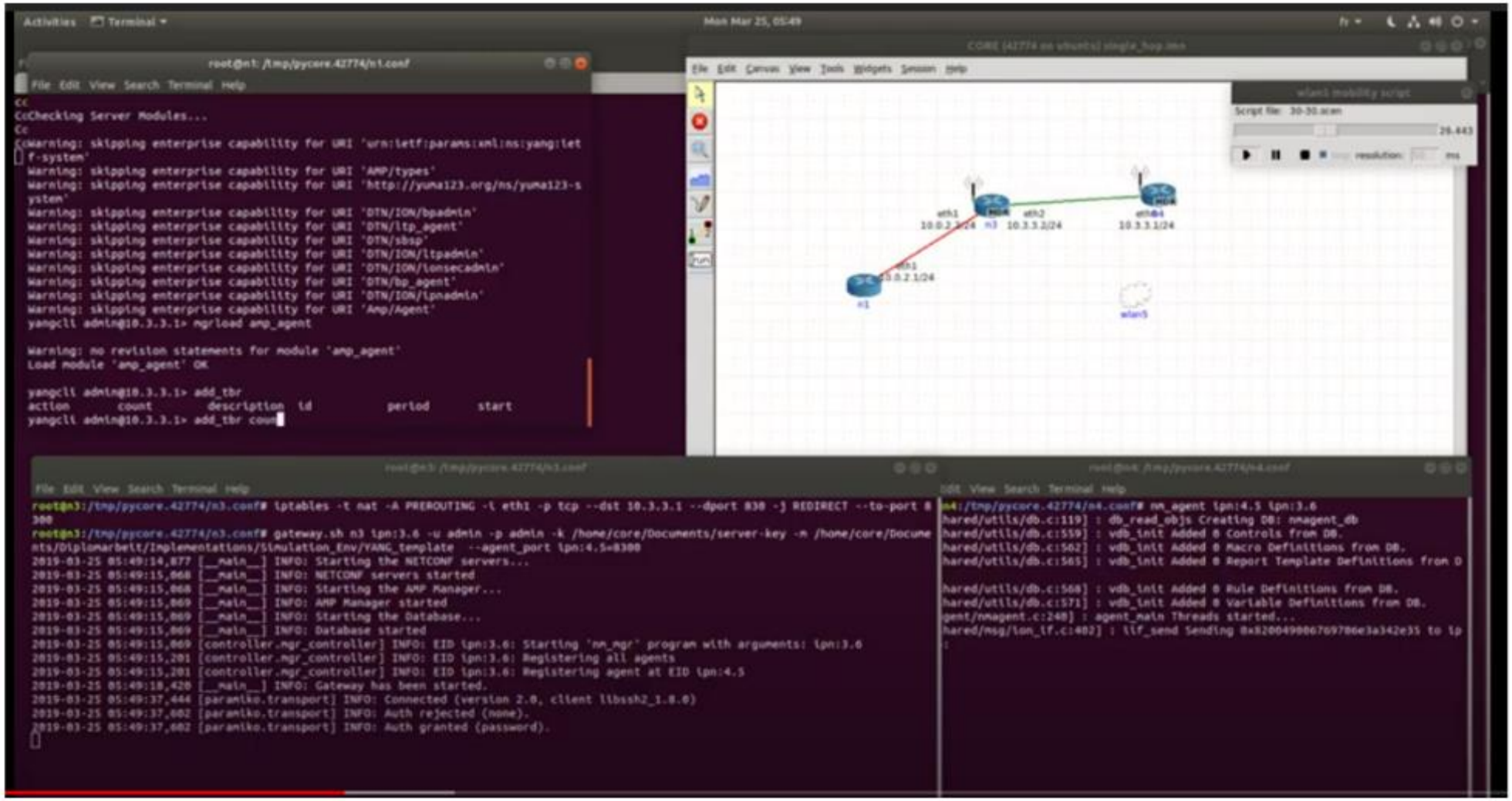
External Work: Integrate with NETCONF



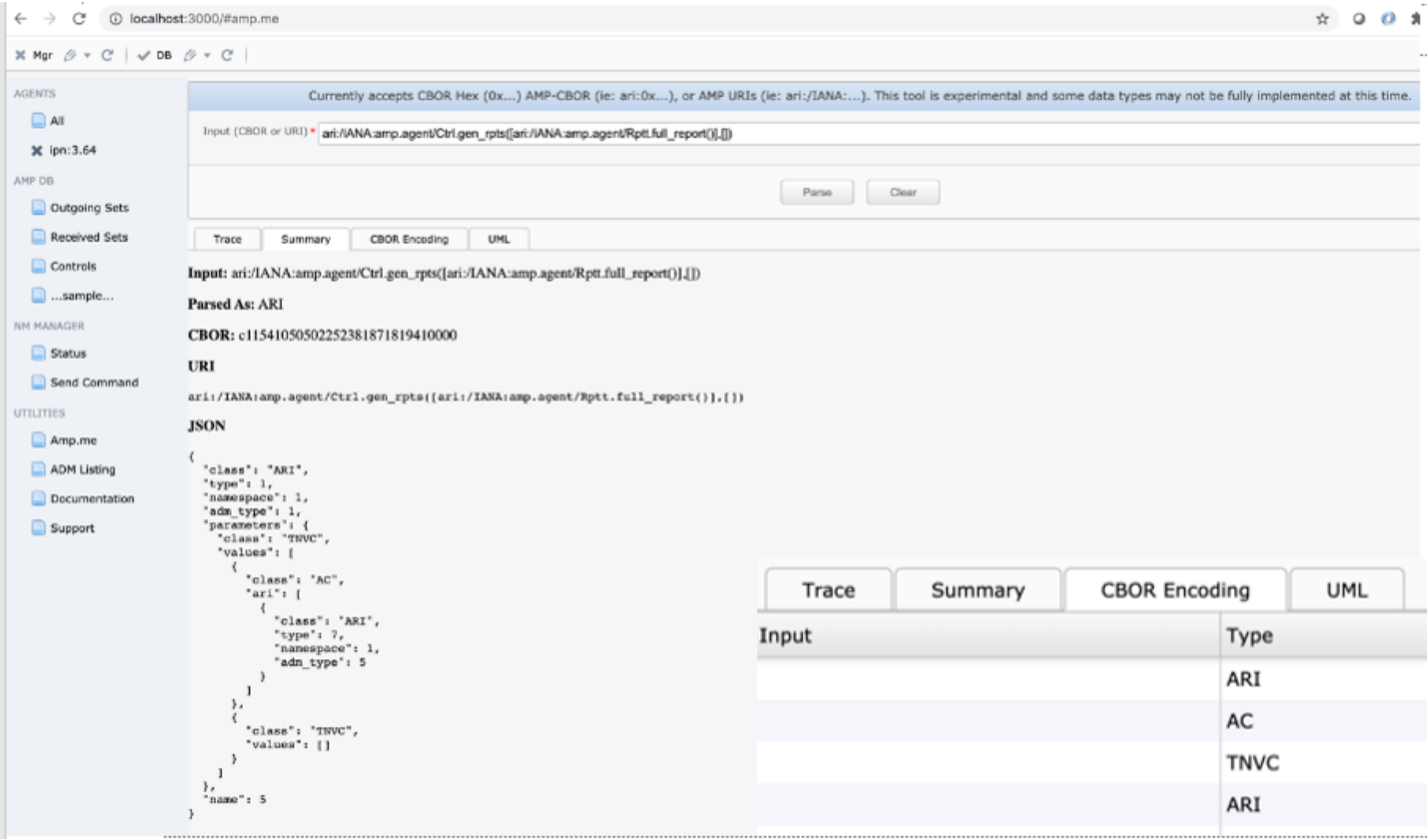
From: Jean Chorin

Live Demo from IETF 104

https://www.youtube.com/watch?v=-AT7mF8Gn94



amp.me – URI Decoding



Trace | Summary | CBOR Encoding | UML

Input	Type	Value
	ARI	ari:/0x8718194100
	AC	ac:/0x818718194100
	TNVC	tnvc:/0x050225238187181941000 0
	ARI	ari:/0xc11541050502252381871819410000 0

amp.me – CBOR Decoding & Re-encoding

Input (CBOR or URI) * ari:/0xc1154105050225238187181941000 0

Parse Clear

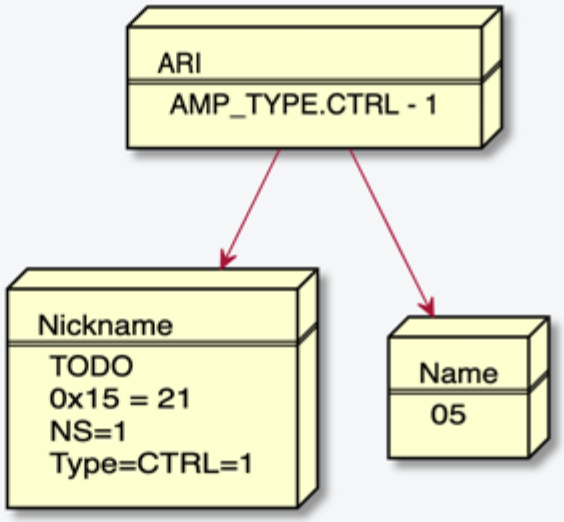
Trace | Summary | CBOR Encoding | UML

Input	Type	Value
	ARI	ari:/0x8718194100
	AC	ac:/0x818718194100
	TNVC	tnvc:/0x050225238187181941000 0
	ARI	ari:/0xc1154105050225238187181941000 0

Trace | Summary | CBOR Encoding | UML

Input	Type	Value
c1	AMP_TYPE.CTRL	NN=true,PARM=true,ISS=false,TAG=false
15	ARI.Nickname	NN=21,NS=1,TYPE=1
4105	ARI.name	[object DataView]
05	TNVC.flag	5
2	TNVC.length	2
25	TNVC.type	AMP_TYPE.AC
23	TNVC.type	AMP_TYPE.TNVC
81	AC.start	1
87	AMP_TYPE.RPTTPL	NN=true,PARM=false,ISS=false,TAG=false
1819	ARI.Nickname	NN=25,NS=1,TYPE=5
4100	ARI.name	[object DataView]
00	TNVC.flag	0
	ARI	ari:/0x8718194100
	AC	ac:/0x818718194100
	TNVC	tnvc:/0x050225238187181941000 0
	ARI	ari:/0xc1154105050225238187181941000 0

Trace | Summary | CBOR Encoding | UML

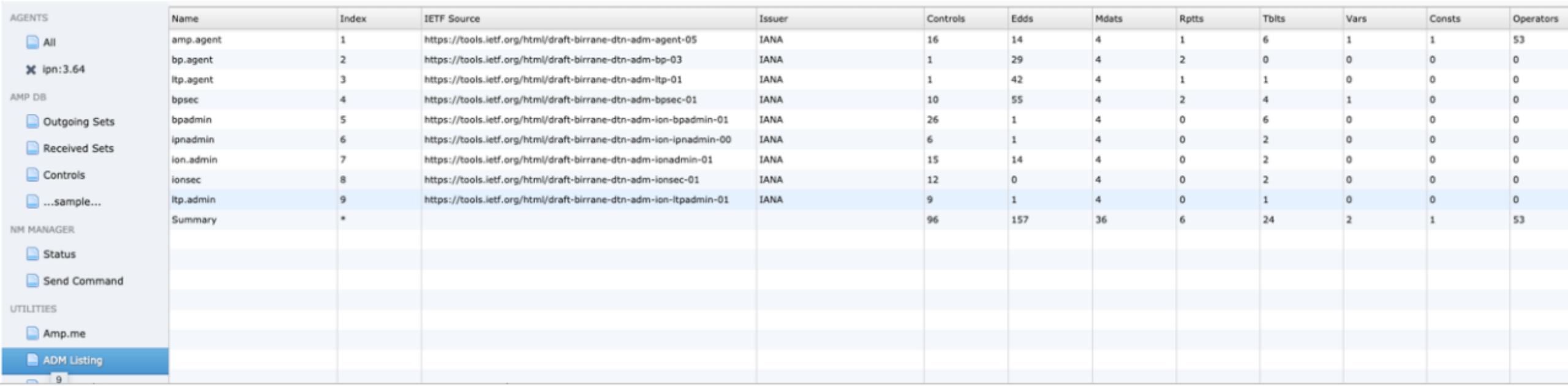


SOL Insertion

```
C02YD01EJG5J-ML:nm_ui edelldj1$ ./sql_msgs_insert.js --msg "ari:/IANA:amp.agent/Ctrl.gen_rpts([ari:/IANA:amp.agent/Rptt.full_rep
ort()],[])" --agent "ipn:3.64"
undefined
[Object: null prototype] {
  ns: 'IANA:amp.agent',
  namespace: 'IANA',
  nsname: 'amp.agent',
  cmd_type: 'Ctrl',
  cmd_name: 'gen_rpts',
  args: '[ari:/IANA:amp.agent/Rptt.full_report()],[]'
}
tnvc fromuri part= [ari:/IANA:amp.agent/Rptt.full_report()]  from uri= [ '[ari:/IANA:amp.agent/Rptt.full_report()]', '[]' ]  typ
e= AC
[Object: null prototype] {
  ns: 'IANA:amp.agent',
  namespace: 'IANA',
  nsname: 'amp.agent',
  cmd_type: 'Rptt',
  cmd_name: 'full_report',
  args: ''
}
tnvc fromuri part= []  from uri= [ '[ari:/IANA:amp.agent/Rptt.full_report()]', '[]' ]  type= TNVC
insertCtrl( 112, null, ari:/IANA:amp.agent/Rptt.full_report()
insertCtrl( 118, 24, ari:/IANA:amp.agent/Ctrl.gen_rpts([ari:/IANA:amp.agent/Rptt.full_report()],[])
Created message group with id of  1
Done
```

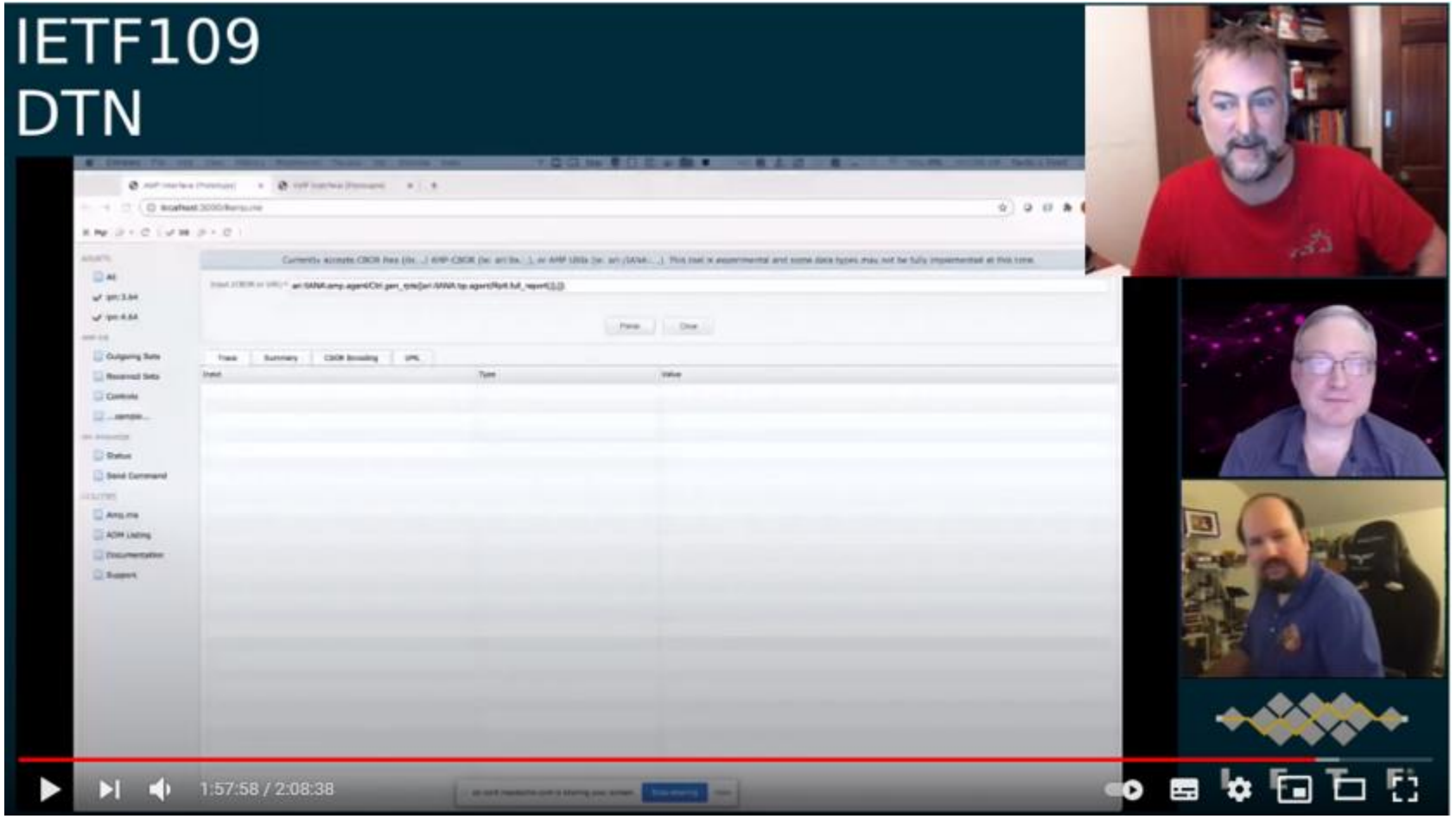
ADMs Listing

- This listing is generated by parsing the ADM JSON files
- Future enhancements would allow viewing details per ADM from the JSON files and/or provide an equivalent view from the MySQL database.



Name	Index	IETF Source	Issuer	Controls	Edds	Mdats	Rptts	Tblts	Vars	Consts	Operators
amp.agent	1	https://tools.ietf.org/html/draft-birrane-dtn-adm-agent-05	IANA	16	14	4	1	6	1	1	53
bp.agent	2	https://tools.ietf.org/html/draft-birrane-dtn-adm-bp-03	IANA	1	29	4	2	0	0	0	0
ltp.agent	3	https://tools.ietf.org/html/draft-birrane-dtn-adm-ltp-01	IANA	1	42	4	1	1	0	0	0
bpsec	4	https://tools.ietf.org/html/draft-birrane-dtn-adm-bpsec-01	IANA	10	55	4	2	4	1	0	0
bpadmin	5	https://tools.ietf.org/html/draft-birrane-dtn-adm-ion-bpadmin-01	IANA	26	1	4	0	6	0	0	0
ipnadmin	6	https://tools.ietf.org/html/draft-birrane-dtn-adm-ion-ipnadmin-00	IANA	6	1	4	0	2	0	0	0
ion.admin	7	https://tools.ietf.org/html/draft-birrane-dtn-adm-ionadmin-01	IANA	15	14	4	0	2	0	0	0
ionsec	8	https://tools.ietf.org/html/draft-birrane-dtn-adm-ionsec-01	IANA	12	0	4	0	2	0	0	0
ltp.admin	9	https://tools.ietf.org/html/draft-birrane-dtn-adm-ion-ltpadmin-01	IANA	9	1	4	0	1	0	0	0
Summary	*			96	157	36	6	24	2	1	53

- https://youtu.be/dNvIdiZklWY?t=7078



JOHNS HOPKINS

APPLIED PHYSICS LABORATORY