

# Asynchronous Management *{Architecture|Template|Protocol}*

***IETF 102***

***Edward Birrane***  
***Edward.Birrane@jhuapl.edu***  
***443-778-7423***



**APL**

JOHNS HOPKINS UNIVERSITY  
**Applied Physics Laboratory**

# Outline

- Overview
- AMA
- ADM Template
- AMP Updates



# Intersection: Spacecraft and Data Center?

## Spacecraft Fault Management Systems

- Stim/Resp Systems
  - Heritage Implementations
  - Deterministic Processing
- 
- Mission Specific Tools
  - Less Infrastructure Funding
  - Network CONOPS

## Terrestrial Network Management - Datacenters

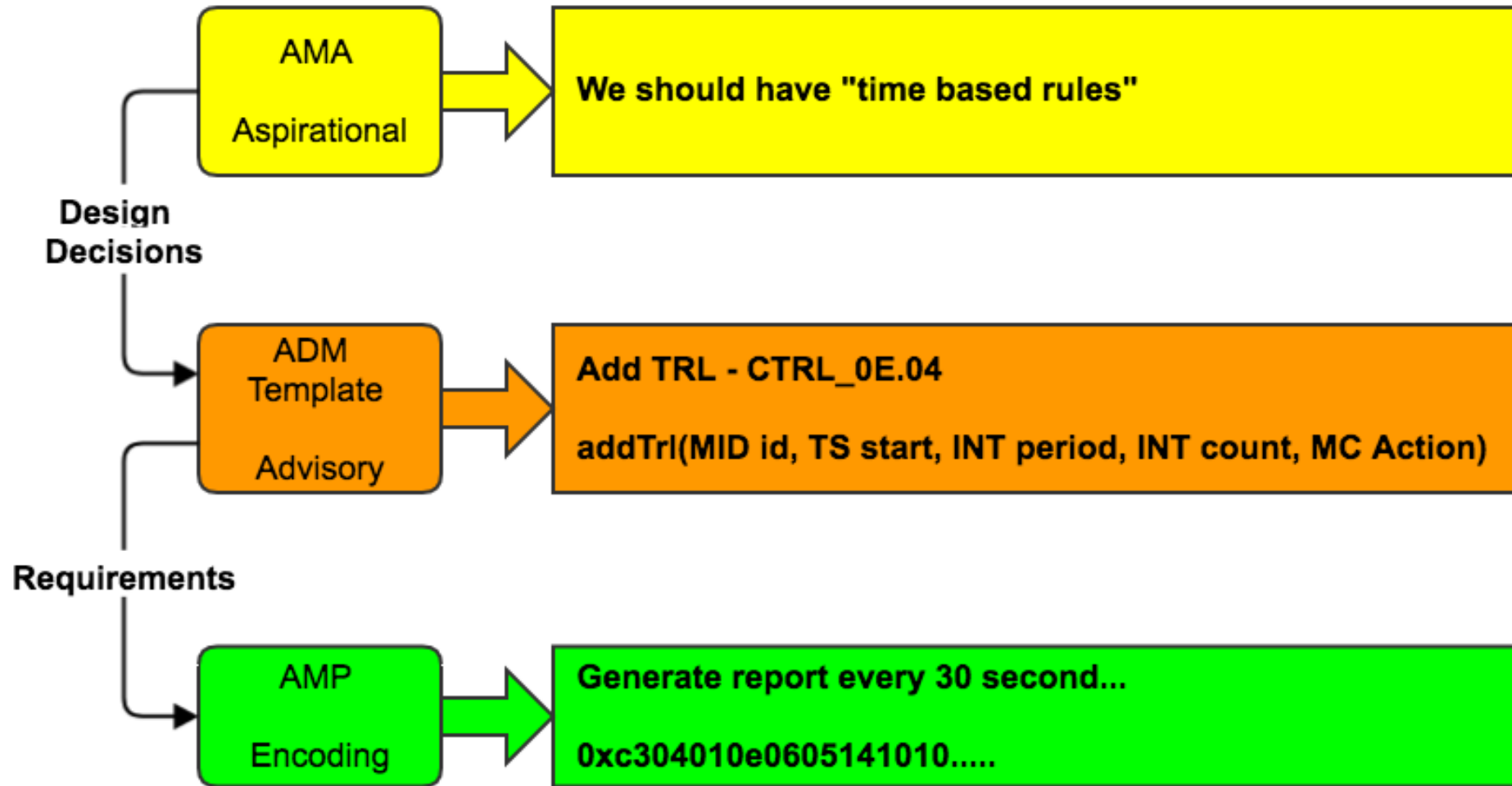
- Lots of Standards
  - Open-source tools
  - Large Investments
- 
- Immature autonomy
  - Inefficient implementations
  - Inefficient protocol layering

What does a healthy intersection of these worlds look like?

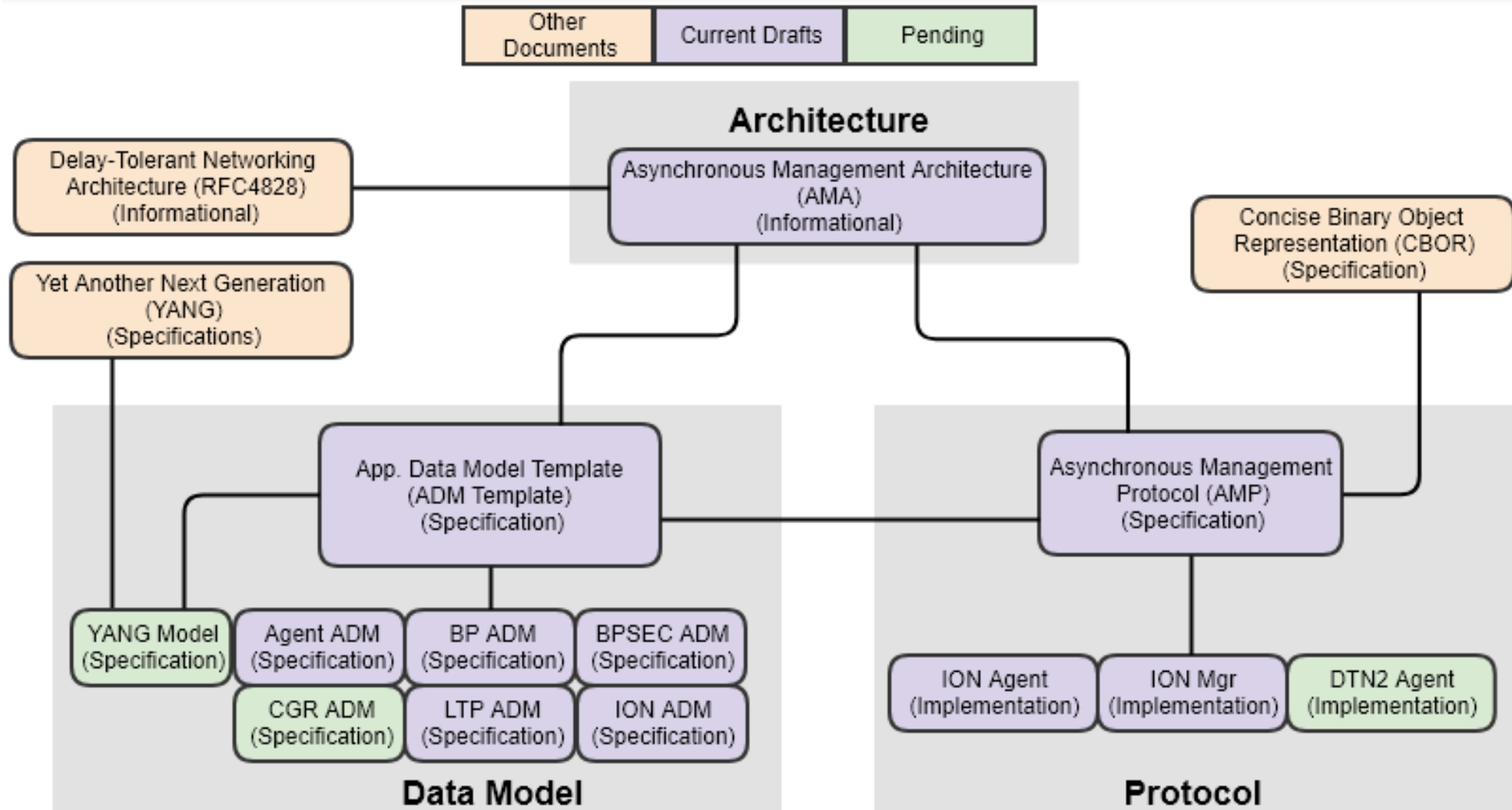
- Automation and Autonomy Model
- Network Management Standards
- Compatibility with commercial work



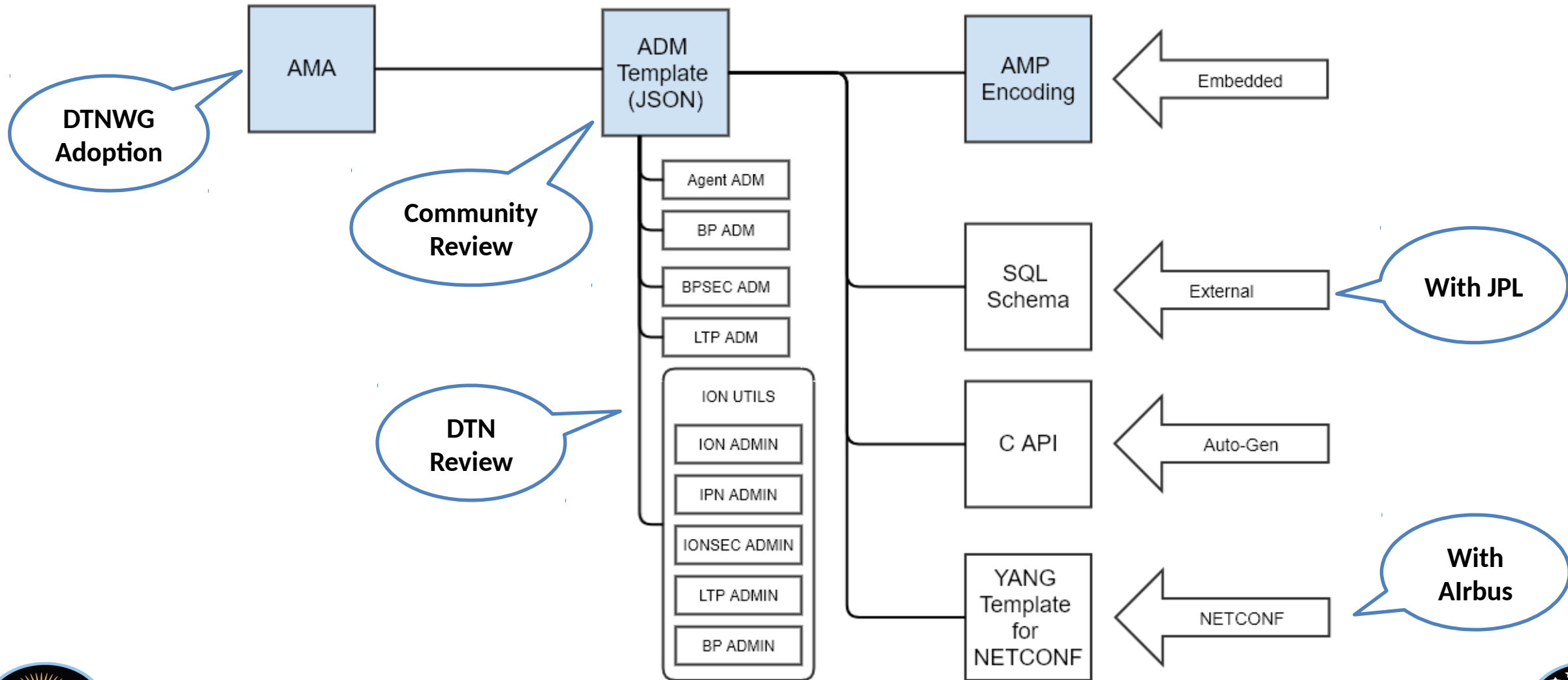
# AMA/ADM/AMP Interactions



# DTN NM Document Tree



# Spec Updates: Tracking ~15 specifications



# 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: Updates

From draft-birrane-dtn-ama-07

- **General**
  - Some minor grammar and spelling.
- **Section 1: Introduction**
  - Minor move of information to scope subsection
- **Section 2: Terminology**
  - Migrated all data model definitions to this document.
- **Section 7: Logical Data Model**
  - Added all data model elements (reports, report templates, tables, table templates, literals)
  - Added some background on command vs control
  - Added sections on time and state based rules.





# AMA: The Simple System Model

From draft-birrane-dtn-ama-07

## ■ Agents

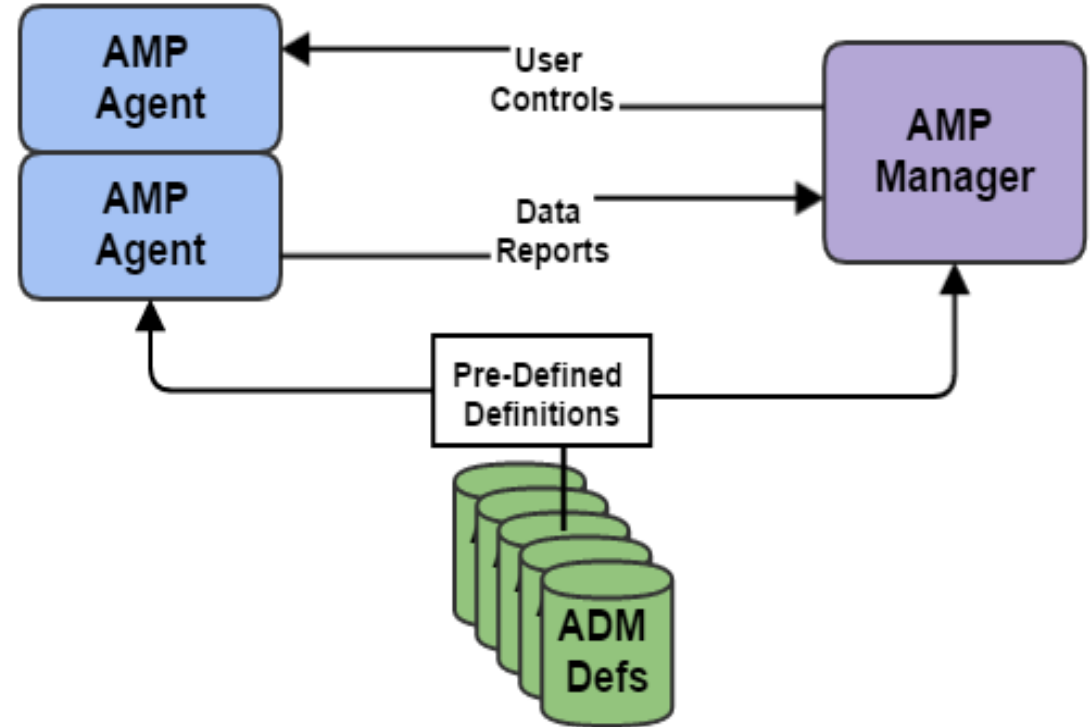
- Run on Managed Devices
- Configure/Report on devices
- Heavy autonomy and parameterized control

## ■ 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 application data models
  - Designed to identify minimum set of information per data model
  - Remove any “encoding hints” from the models.
- 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)

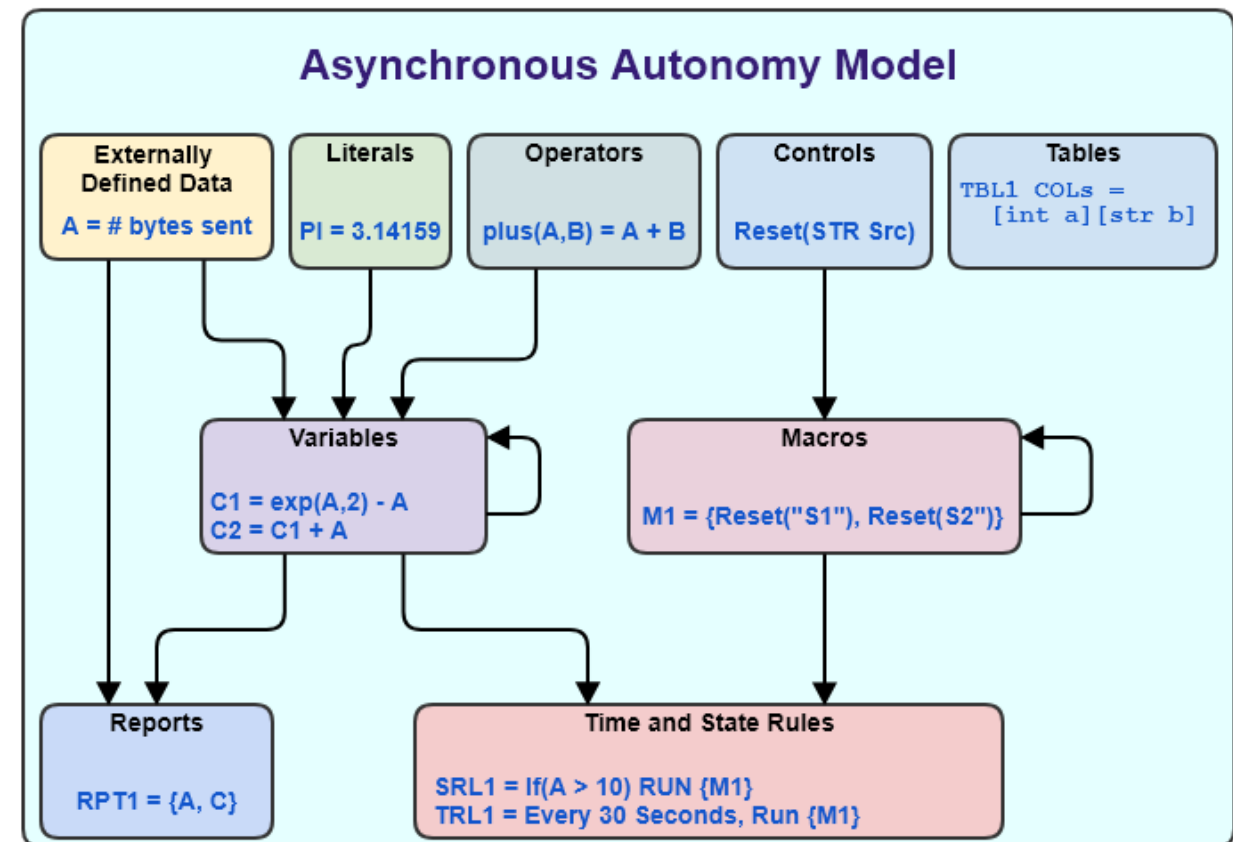
## “Atomic” Elements

- **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.



# Model Relationships in the ADM

Data Model Relationship

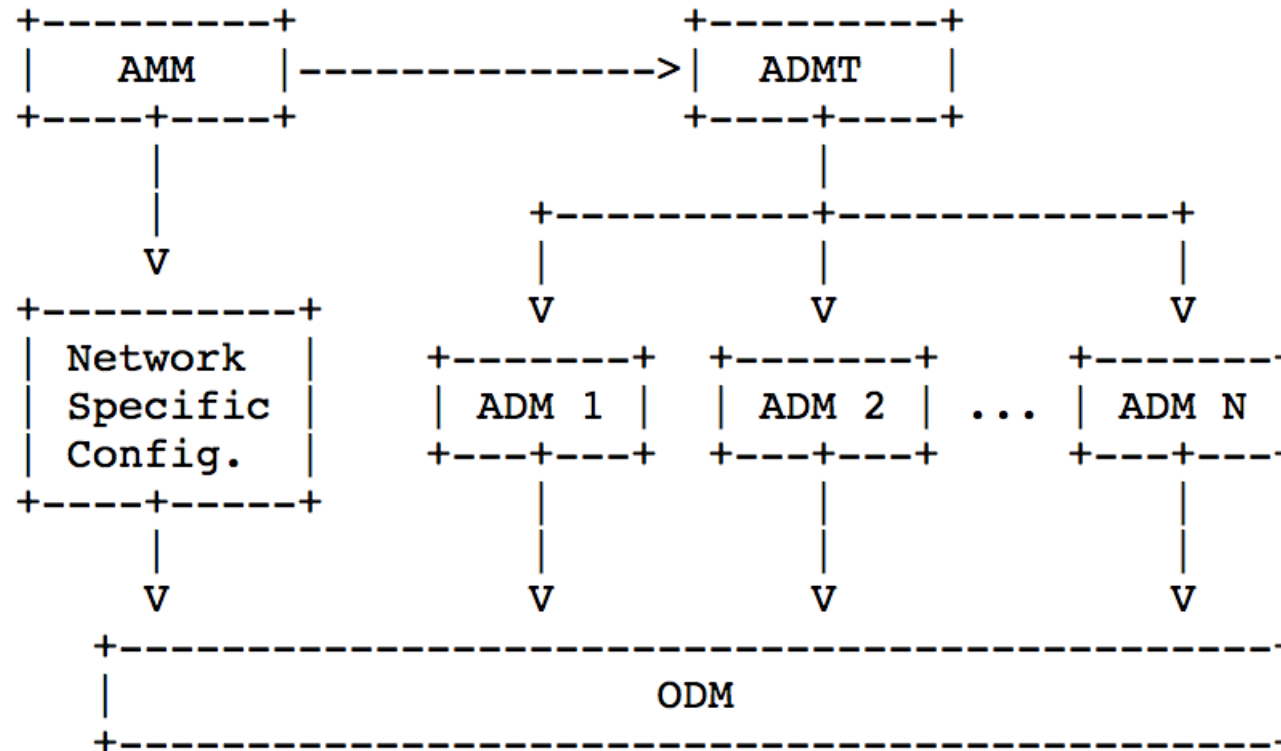


Figure 1

# 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":"cname"}],  
    "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.”
```

# Brief AMP Status Update

- AMP is a binary encoding and compression of ADM information.
  - Also, a set of messages to exchange these encodings
  - Also, the state machines to discuss how ADM information is updated
  - Recently updated to migrate to CBOR representation.
- AMP Implementation Status
  - Reference implementation in the DTN ION suite being updated to most recent version of AMP.
  - New implementation for CFS should be starting
- Other AMP-Related Activity
  - AMP Wireshark plugin completed. Needs to be updated to new CBOR encoding
  - CAMP: Python scripts to auto-generate ION C code and SQL statements for schema







# 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;
}
```

