TAXII

TAXII is an open protocol for the communication of cyber threat information. Focusing on simplicity and scalability, TAXII enables authenticated and secure communication of cyber threat information across products and organizations.

MVP

TAXII SC Status Report

- Identified TAXII 1.0/1.1 Pain Points
- Created a strawman for TAXII 2.0
- Resolved multiple items within the SC
- Started work on a TAXII 2.0 pre-draft spec

TAXII 1.0 / 1.1 Pain Points

- Too much optionality (Services, Authentication, etc)
- Inefficient communication patterns
- Lack of automated discovery
- Envelope/Content are not easily separable
- Reinvented some HTTP functionality
- XML

https://github.com/TAXIIProject/TAXII-Specifications/wiki/TAXII-1.1-1.0-Pain-Points

Resolved within SC

- Crafted a vision statement
- Strawman of TAXII 2.0
- Supporting multiple protocols no longer a design requirement
  - Defined HTTPS as MTI
- Settled on a RESTful design pattern
- Decided that there must be an authentication MTI
- Began work on a pre-draft specification document

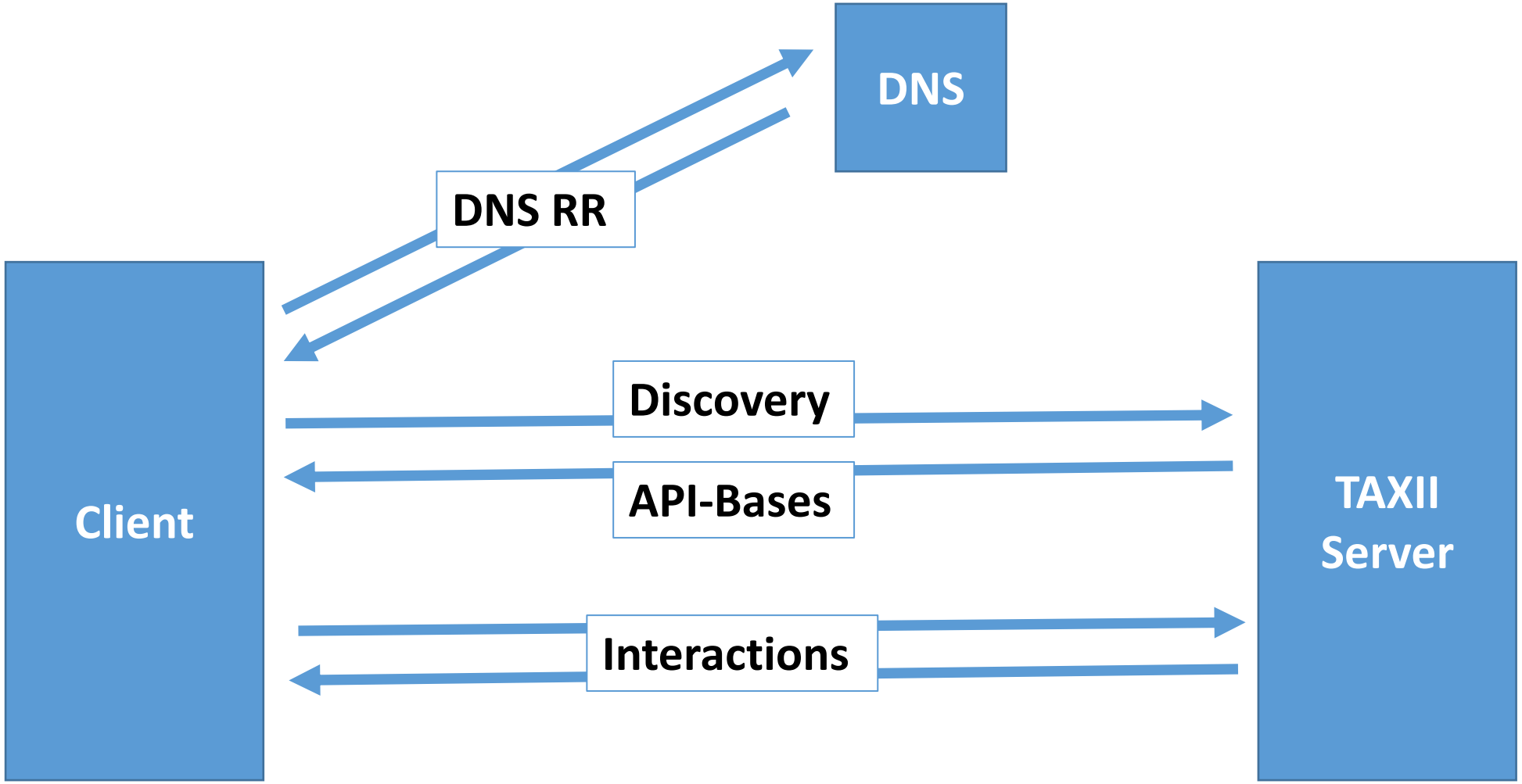
TAXII 1.x to 2.0

- **Discovery** still exists
- **Collections** elevated and formalized as a “container of content”
- TAXII 1.x Services factored out as HTTP Verbs
  - Inbox = POST/PUT to a Collection
  - Poll = GET from a Collection
  - List Collections = GET
  - Discovery = GET
- **Channels** added and formalized a Publish/Subscribe pattern
- Decided Authentication MTI is required
- Feature parity from 2.0 to 1.x

Authentication

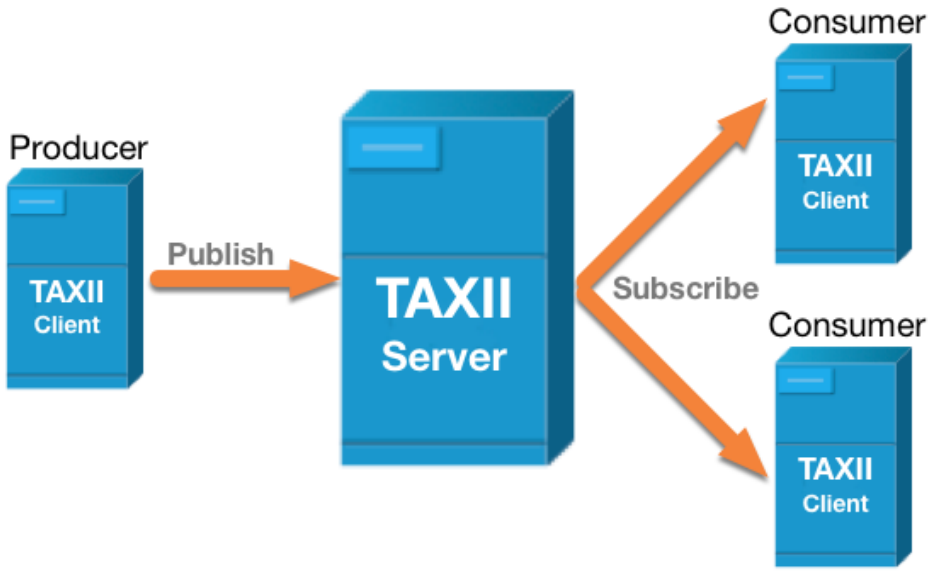
- Need an Authentication MTI
  - Other Auth methods will be allowed, but not required
- What should we do?
- We have heard that the base MTI authentication should be HTTP BASIC
- What about JWT?
- What about extensions?

The Ugly but informative(?) drawing

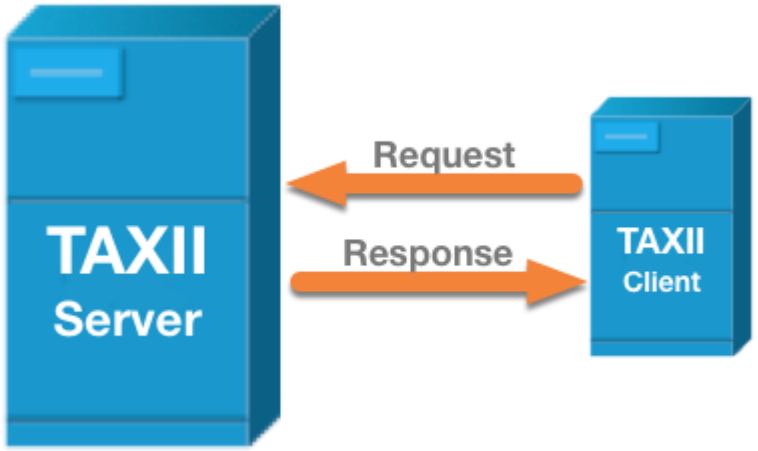


Where are we now?

Channels – A mechanism for moving messages from producer to consumer



Collections – A grouping of threat information



API Summary

Path	GET	POST	DELETE
[Discovery]	Server Info + List API-Bases	n/a	n/a
[API-BASE]/[?p1=v1&p2=v2]	Global Search	n/a	n/a
[API-BASE]/collections/	List Collections	Create Collection	n/a
[API-BASE]/channels/	List Channels	Create Channel	n/a

Collections Detail

Path [API-Base]/	GET	POST	DELETE
/collections/	List Collections	Create Collection	n/a
/collections/<name>/[?p1=v1&p2=v2]	List/Search objects in Collection	Create Object	Delete Collection
/collections/<name>/<type>/[?p1=v1&p2=v2]	List/Search all objects of <type>	Create an object of <type>	Delete all objects of <type>
/collections/<name>/<type>/<id>/	Get object with ID=<id>	Update object with ID=<id>	Delete object with ID=<id>

Channels Detail

Path **[API-Base]/**	GET	POST	DELETE
/channels/[?p1=v1&p2=v2]	List/Search Channels	Create Channel	n/a
/channels/<name>/[?p1=v1&p2=v2]	Get Message(s) (Filter optional)	Send Message(s)	Delete Channel

Example #1 – Discovery

Request	Response

GET [Discovery]	{
	"type": "discovery",
	"server-description": "My Cool Server",
	"server-icon": "https://example.com/favicon.ico",
	"api-bases": [
	{
	"url": "https://example.com/taxii/api-base-1/",
	"description": "Public TAXII Interface"
	},
	{
	"url": "https://example.com/taxii/api-base-2/",
	"description": "Private TAXII Interface"
	}
	]
	}

Example #1 – GET Indicator (STIX 1.x content)

Request	Response
GET [API-Base]/collections/indicator/1234	HTTP 200 OK
Accept: STIX-2.0, q=0.9; STIX-1.2.1, q=0.8	Content-Type: STIX-1.2.1
Content:	<xml>Some stuff</xml>

TAXII Query

- Trey’s Proposal
- How should we do this?
  - Is a REST thing sufficient for MVP, or do we need more?
- What should it look like?

Open Discussion

- Should TAXII **require** that servers implement Channels and Collections?
- How should TAXII handle RBAC?
- MVP & Dates

Questions?