



AsyncAPI

AsyncAPI Specs from Real-World API Data



Rob Galanakis

Cofounder, Principal, WebhookDB

 [@robgalanakis](https://twitter.com/robgalanakis)



Natalie Edson

Senior Engineer, WebhookDB

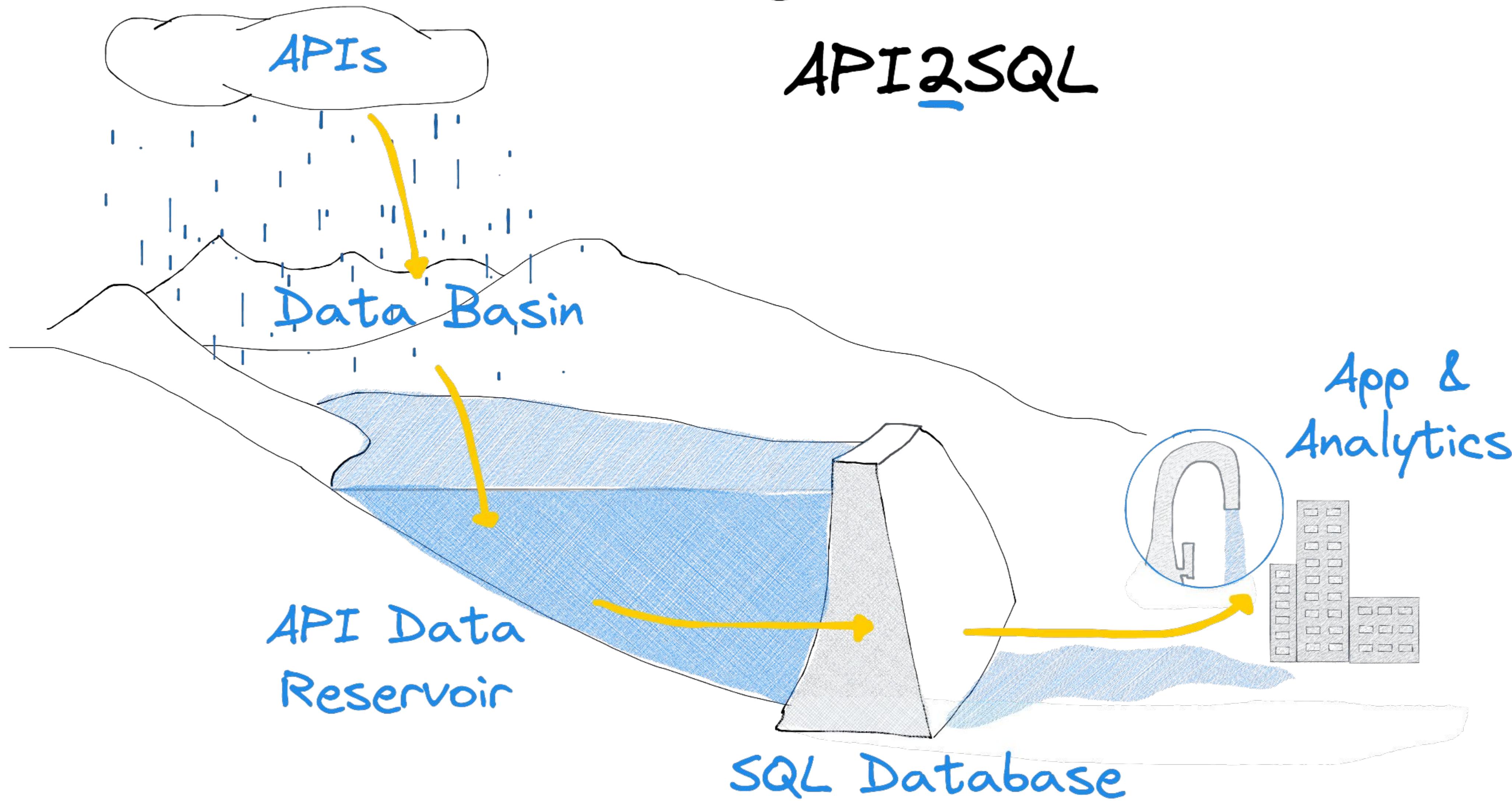


“

An application is any kind of computer program or a group of them. It MUST be a producer, a consumer or both. An application MAY be a microservice, IoT device (sensor), mainframe process, etc. An application MAY be written in any number of different programming languages as long as they support the selected protocol. An application MUST also use a protocol supported by the server in order to connect and exchange messages.

WebhookDB

API2SQL

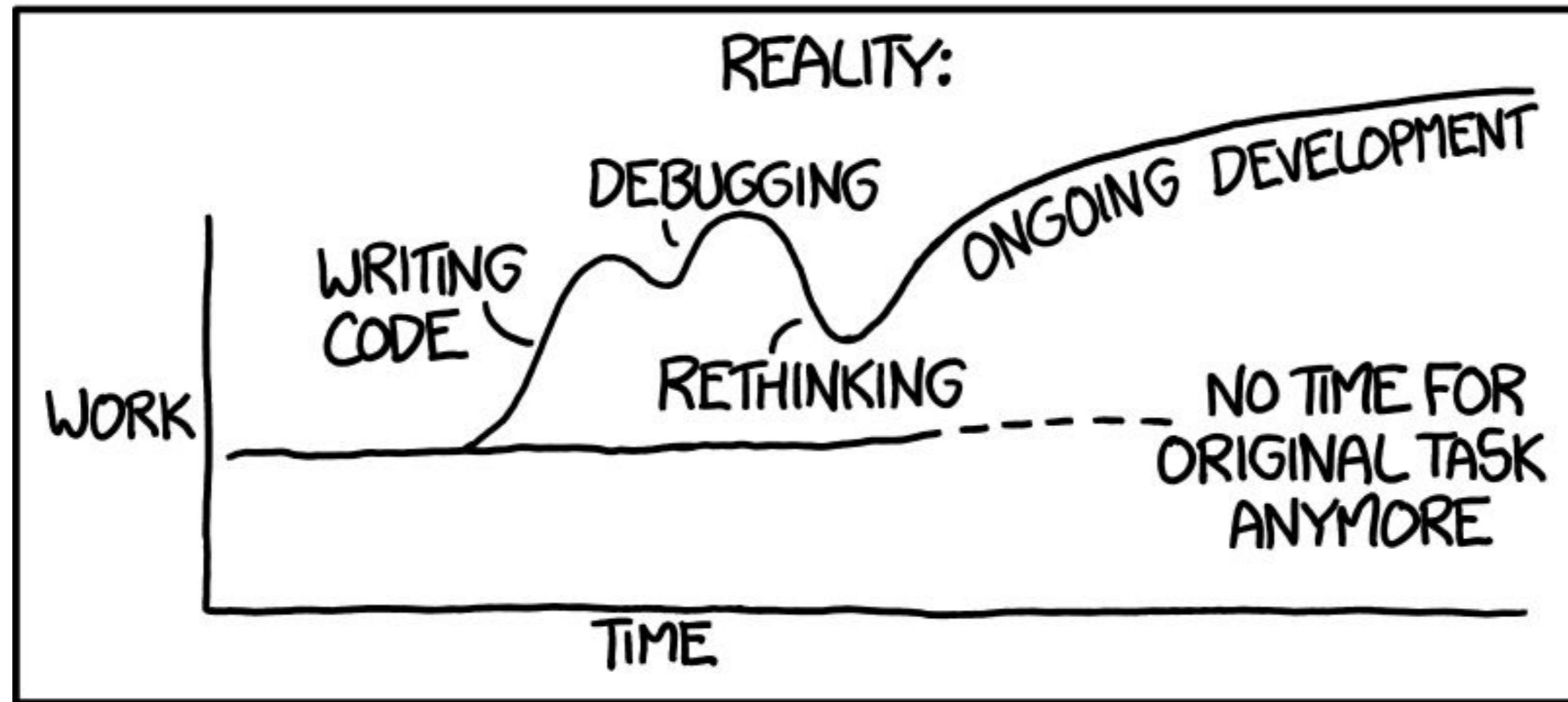
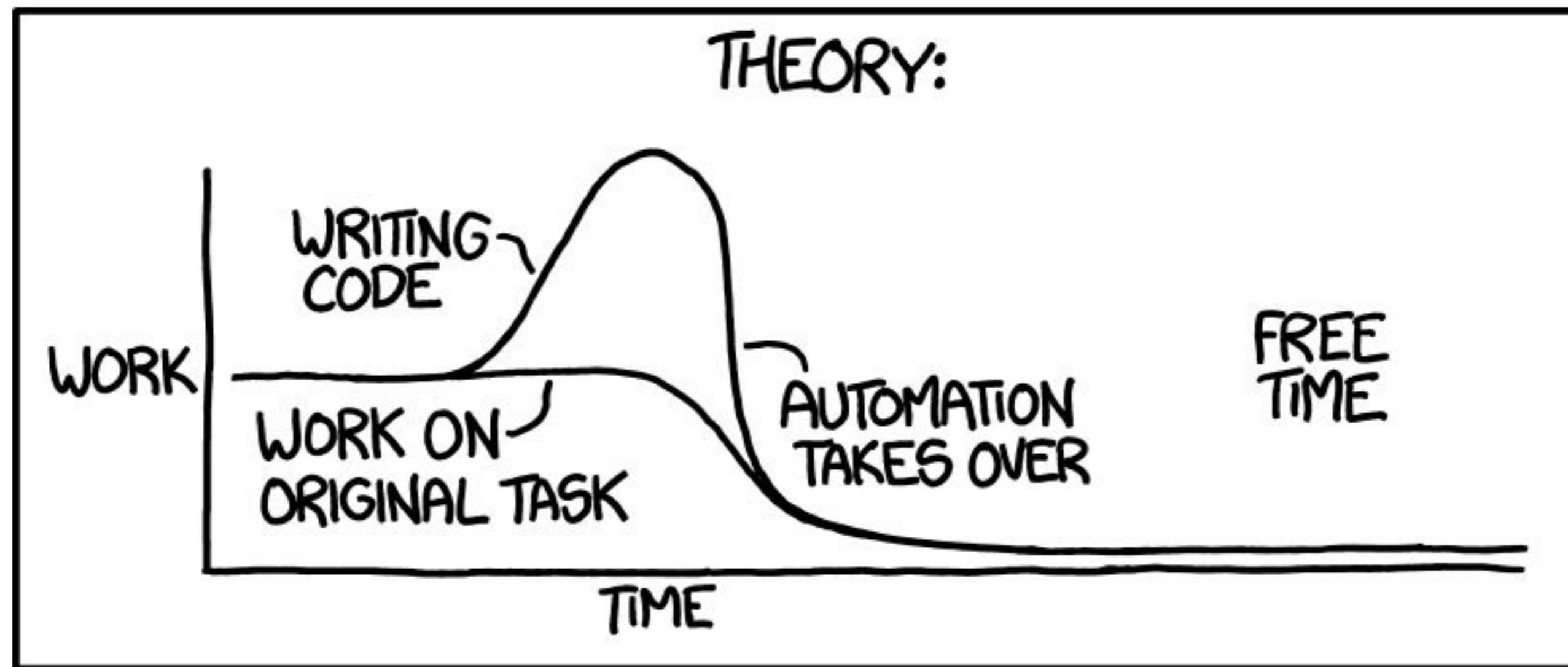


**API Integration is like a box of chocolates.
You never know what you're gonna get.**

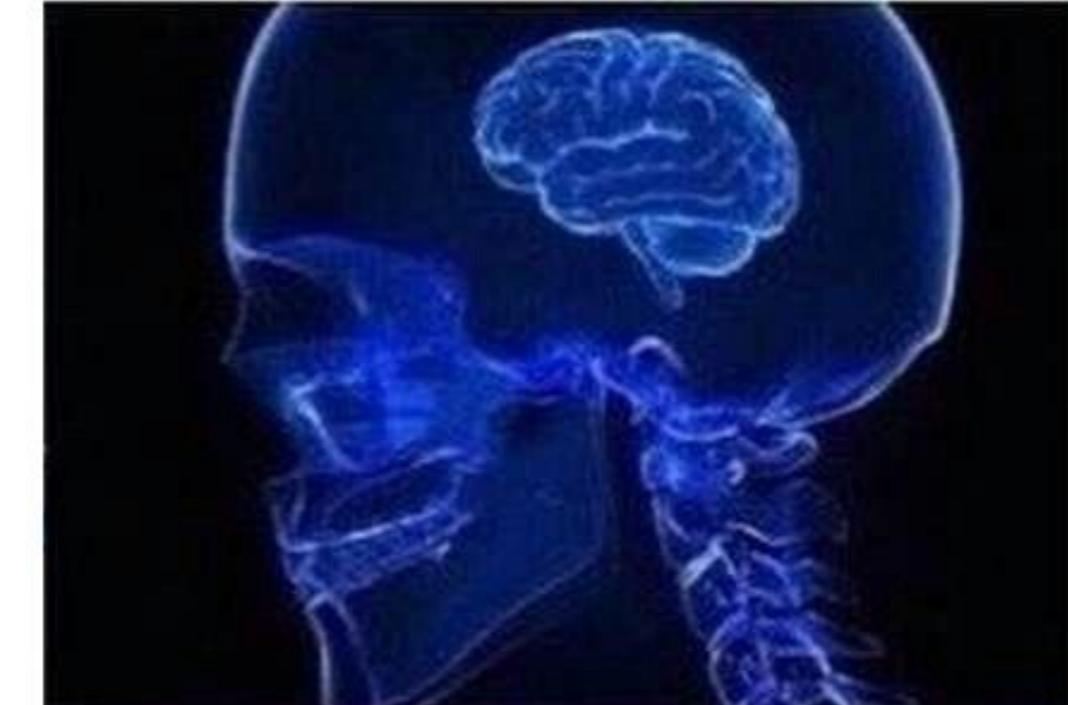


```
{  
  "currentUserHasAccessToDetails": true,  
  "DeletedDate": null,  
  "end_date": "06/10/2022 12:30",  
  "start_date": "06/10/2022 11:30",  
  "LastKeptAppointmentDateTimeMDY": "05/02/2022 10:00 AM",  
  "Age": "15",  
  "NextAppointmentDateTimeMDY": "-",  
  "status": {  
    "AbsenceReason": null,  
    "Status": "Upcoming"  
  },  
  "statusBadge": "",  
  "statusJson": "{\"Status\":\"Upcoming\",\"AbsenceReason\":null}",  
  "teletherapySessionId": null,  
  "teletherapyTitle": null,  
  "type": 1  
}
```

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



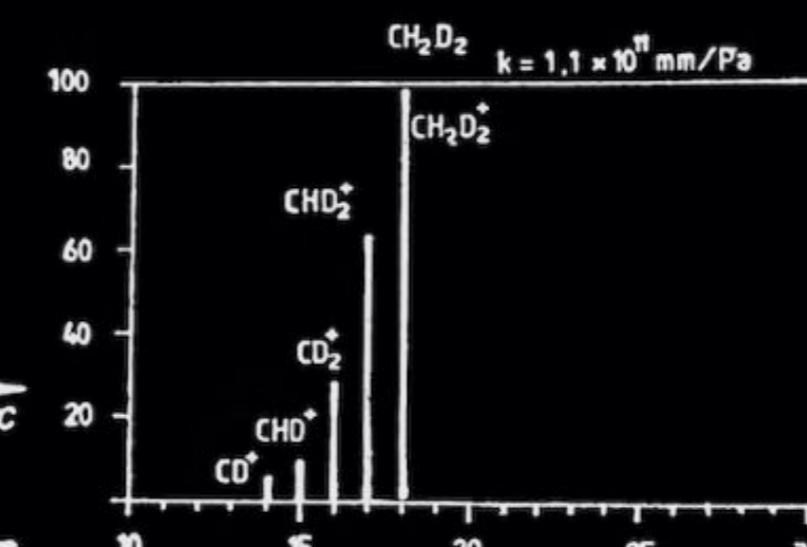
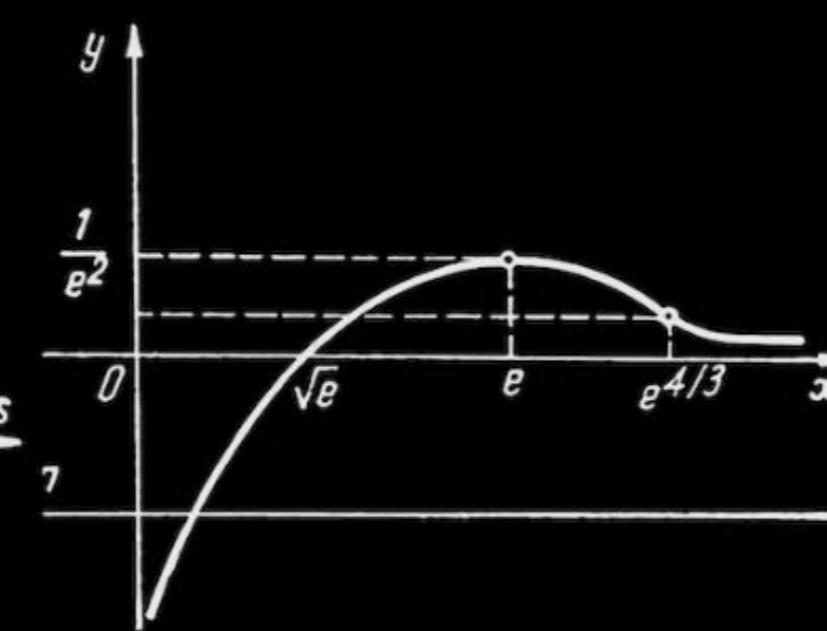
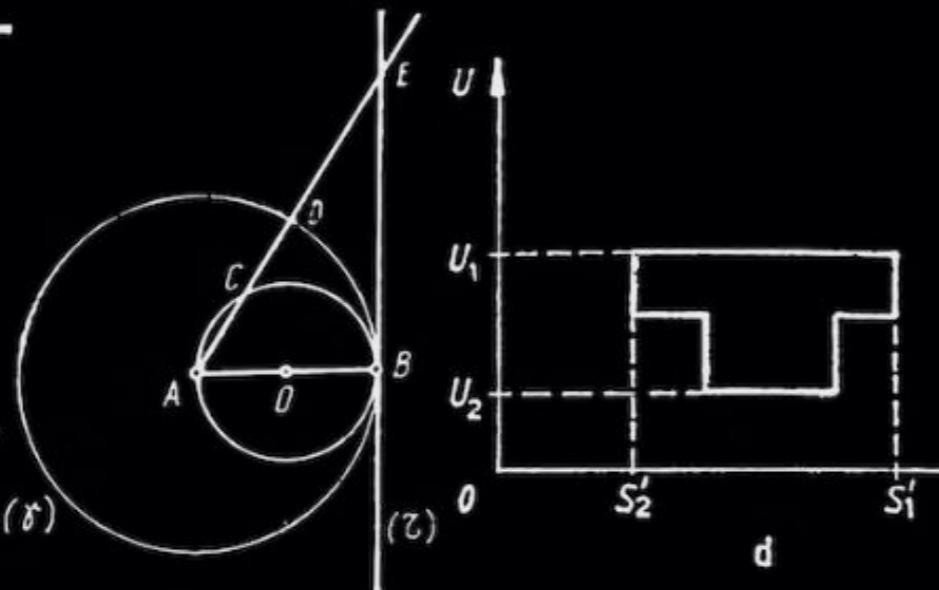
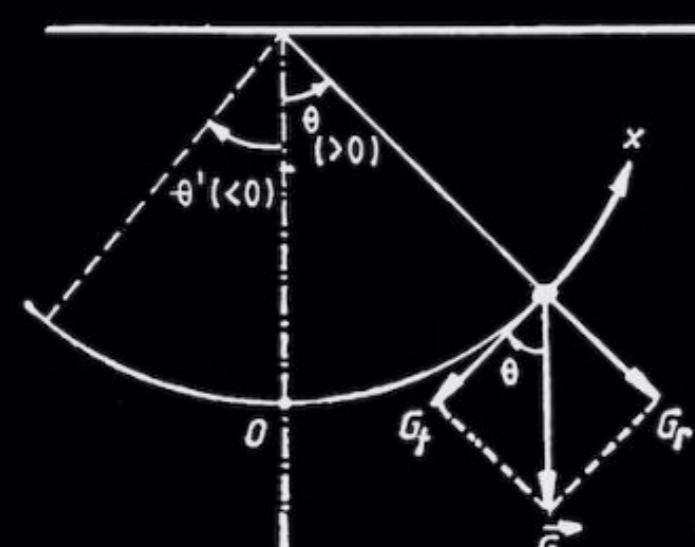
Read docs to
integrate API.



“

Instead of circumscribing development, specifications can be used to explore unfamiliar systems and navigate how we integrate with them.



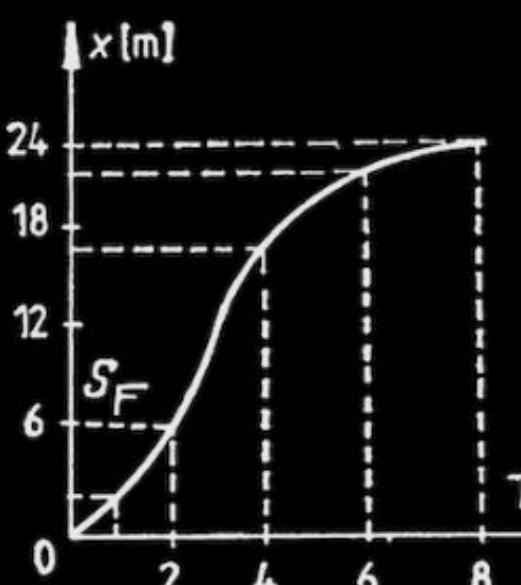
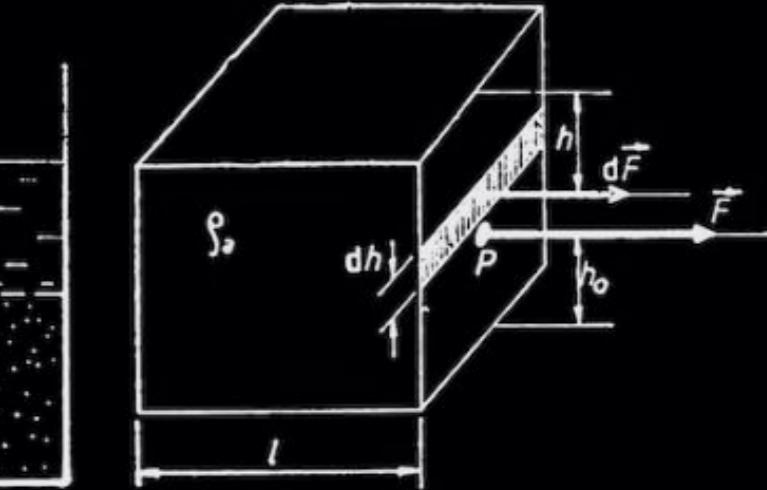


$$\begin{aligned}
 E_p &= E_{p_{\max}} \Rightarrow \sin^2 \left(3t_p + \frac{\pi}{3} \right) = 1 \\
 &= \sin \left(\frac{\pi}{2} + n\pi \right); n = 0, 1, 2, \dots \\
 t_p &= \frac{\pi}{3} \left(n + \frac{1}{6} \right); n = 0, 1, 2, \dots \\
 E_c &= E_{c_{\max}} \Rightarrow \cos^2 \left(3t_c + \frac{\pi}{3} \right) = 1 \Rightarrow \cos \left(3t_c + \frac{\pi}{3} \right) = \pm 1 = \cos(n\pi) \Rightarrow t_c = \frac{\pi}{3} \left(n - \frac{1}{3} \right)
 \end{aligned}$$

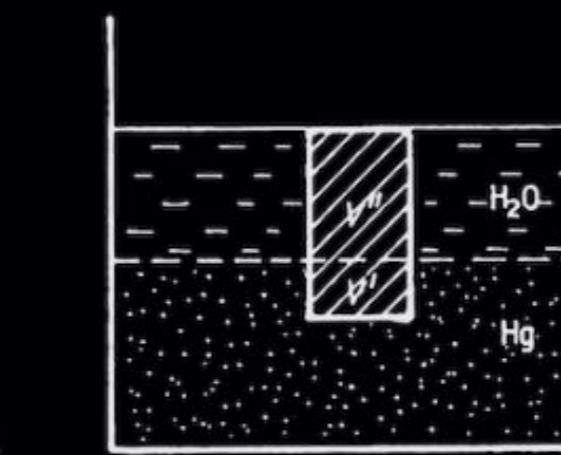
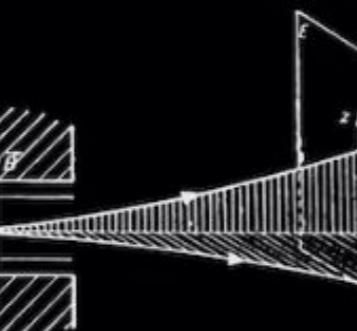
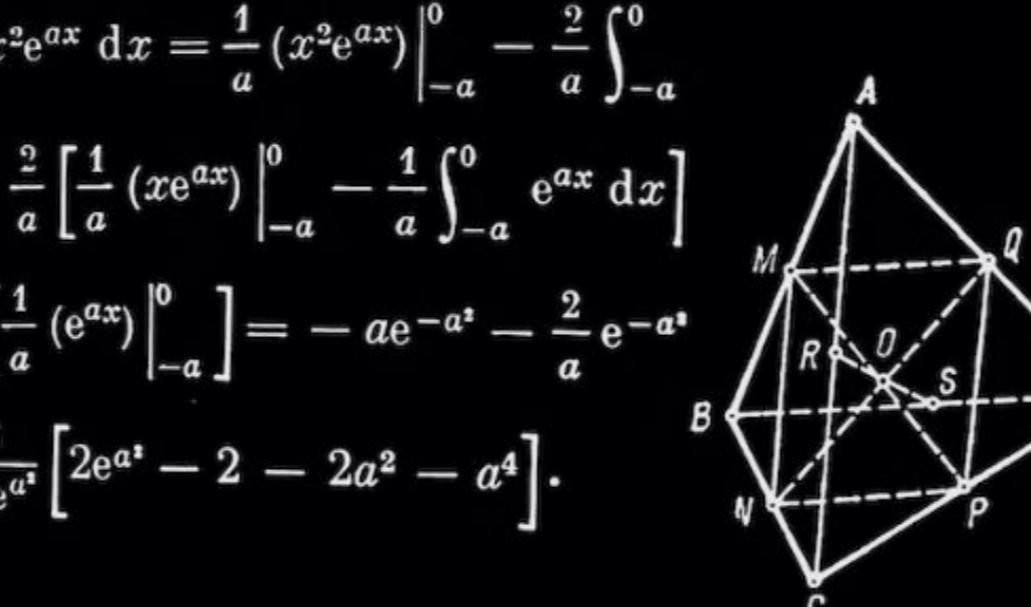
$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{4\pi m_1 K \rho}{3m_1}} = \sqrt{\frac{4\pi K \rho}{3}}$$

$$\omega = \sqrt{\frac{g_0}{R_0}},$$

$$T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{R_0}{g}} = 5,03 \cdot 10^3 \text{ s}.$$



$$\begin{aligned}
 \frac{1 - \left(-\frac{1}{n+2} \right)^{n+1}}{1 + \frac{1}{n+2}} + \frac{1}{n+1} \cdot \frac{1 - \left(-\frac{1}{n+1} \right)^{n+1}}{1 + \frac{1}{n+1}} - a^2 - \frac{2}{a} \left[\frac{1}{a} (xe^{ax}) \Big|_0^0 - \frac{1}{a} \int_{-a}^0 e^{ax} dx \right] \\
 - \frac{1}{n+1}^{n+1} - \frac{1 - \left(-\frac{1}{n+2} \right)^{n+1}}{n+3} = + \frac{2}{a^2} \left[\frac{1}{a} (e^{ax}) \Big|_{-a}^0 \right] = -ae^{-a^2} - \frac{2}{a} e^{-a^2} \\
 (-1)^{n+1} \frac{1}{(n+2)^n} + (-1)^n \cdot \frac{n+3}{n+1} \cdot \frac{1}{(n+1)^{n+1}} = = \frac{1}{a^3 e^{a^2}} \left[2e^{a^2} - 2 - 2a^2 - a^4 \right].
 \end{aligned}$$



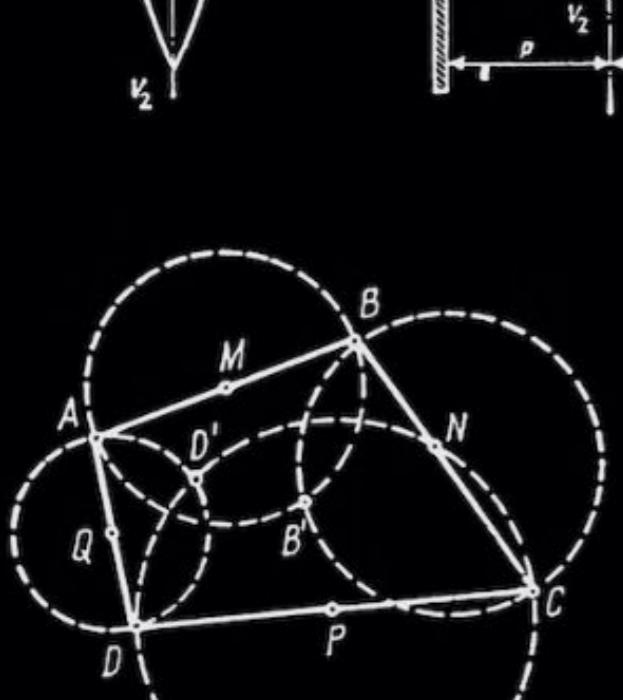
$$Q_{\text{total}} = Q_1 + Q_2 = 3\epsilon_0 \frac{S}{d_1} U_0$$

$$C_1 = C_2 = \epsilon_0 \frac{S}{d_1} = 8,85 \text{ pF}$$

$$Q = \frac{Q_1 + Q_2}{2} = 13,275 \cdot 10^{-9} \text{ C}$$

$$U = \frac{Q}{C_1} = \frac{3}{2} U_0 = 1500 \text{ V}$$

$$= \frac{1}{2} Q U = \frac{9}{8} \epsilon_0 \frac{S}{d_1} U_0^2 = 9,956 \cdot 10^{-6} \text{ J}$$



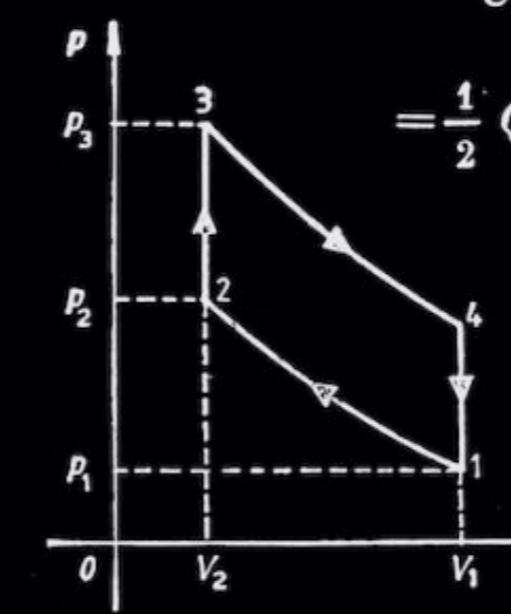
$$\begin{aligned}
 Q_{41} &= vCT_1(1 - \epsilon^{1/2}) + vC_V T_1(\mathcal{K} - 1), \\
 Q_{34} &= vC_V T_2(\mathcal{K} - 1) + vCT_4(1 - \epsilon^{1/2}), \\
 \frac{1}{T_2} &= \mathcal{K}, \quad \frac{T_3}{T_4} = \epsilon^{1/2}, \quad \frac{T_4}{T_1} = \mathcal{K}, \\
 \frac{T_3}{T_2} &= \mathcal{K}, \quad \frac{T_3}{T_4} = \epsilon^{1/2}, \quad \frac{T_4}{T_1} = \mathcal{K}
 \end{aligned}$$

$I[\text{mA}]$	0	0	4	50	104	170
$U[\text{V}]$	0	0,5	0,6	0,8	0,9	1,0
$I[\text{mA}]$	0	-1,05	-2,1	-3,2	-4,2	-5,3
$U[\text{V}]$	0	-1	-2	-3	-4	-5
$I[\text{mA}]$	0	0	4	44	115	175
$U[\text{V}]$	0	0,4	0,6	0,8	0,9	1,0
$I[\text{mA}]$	0	-0,4	-0,76	-1,12	-1,5	-1,9
$U[\text{V}]$	0	-1	-2	-3	-4	-5
$I[\text{mA}]$	0	1,4	2,8	4,2	5,6	7,1
$U[\text{V}]$	0	1	2	3	4	5
$I[\text{mA}]$	0	-1,4	-2,8	-4,2	-5,6	-7,1
$U[\text{V}]$	0	-1	-2	-3	-4	-5

$$-(x + t)I_2 + (xt - yz)I_2 = 0.$$

$$\begin{pmatrix} x & y \\ z & t \end{pmatrix} - \begin{pmatrix} x + t & 0 \\ 0 & x + t \end{pmatrix} = \begin{pmatrix} -t & y \\ z & -x \end{pmatrix}.$$

$$\begin{pmatrix} y & -t & 0 \\ t & z & -x \end{pmatrix} = \begin{pmatrix} yz - xt & 0 \\ 0 & yz - tx \end{pmatrix} = \\
 yz - xt)I_2 = -(xt - yz)I_2,$$

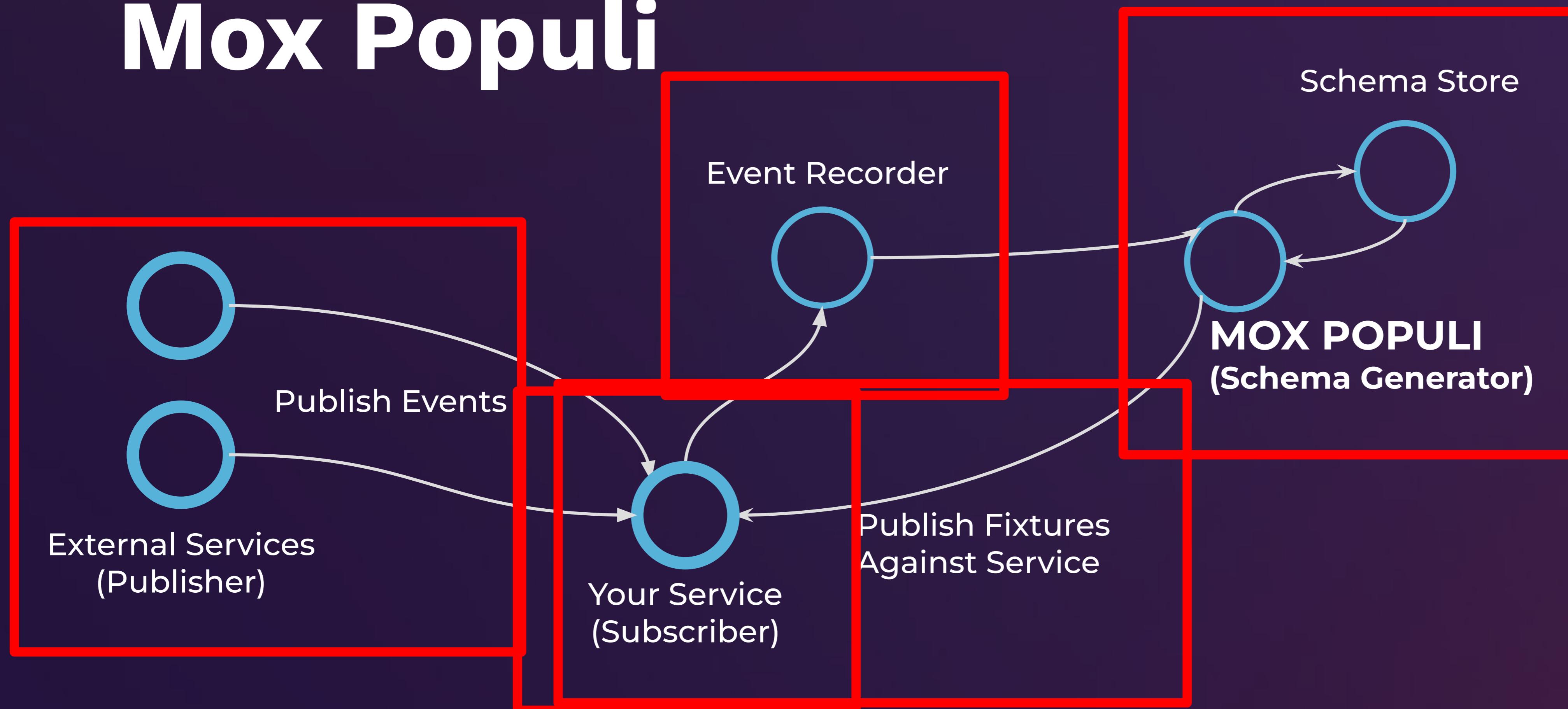


$$\begin{aligned}
 -Q_{41} &= vCT_1(1 - \epsilon^{1/2}) + vC_V T_1(\mathcal{K} - 1), \\
 -Q_{34} &= vC_V T_2(\mathcal{K} - 1) + vCT_4(1 - \epsilon^{1/2}), \\
 \frac{1}{T_2} &= \mathcal{K}, \quad \frac{T_3}{T_4} = \epsilon^{1/2}, \quad \frac{T_4}{T_1} = \mathcal{K}, \\
 \frac{T_3}{T_2} &= \mathcal{K}, \quad \frac{T_3}{T_4} = \epsilon^{1/2}, \quad \frac{T_4}{T_1} = \mathcal{K}
 \end{aligned}$$

SPECIFICATIONS FOR THE PEOPLE
MOX POPULI

SPECIFICATIONS FOR THE PEOPLE

Mox Populi



SPECIFICATIONS FOR THE PEOPLE

Mox Populi

- 1. Recording Events
- 2. Generating JSONSchema for Message Payloads
- 3. Generating the rest of AsyncAPI Specification
- 4. Fixturing to test your integration

STEP ONE

Recording Events

Sample and record full-fidelity event data into a database.

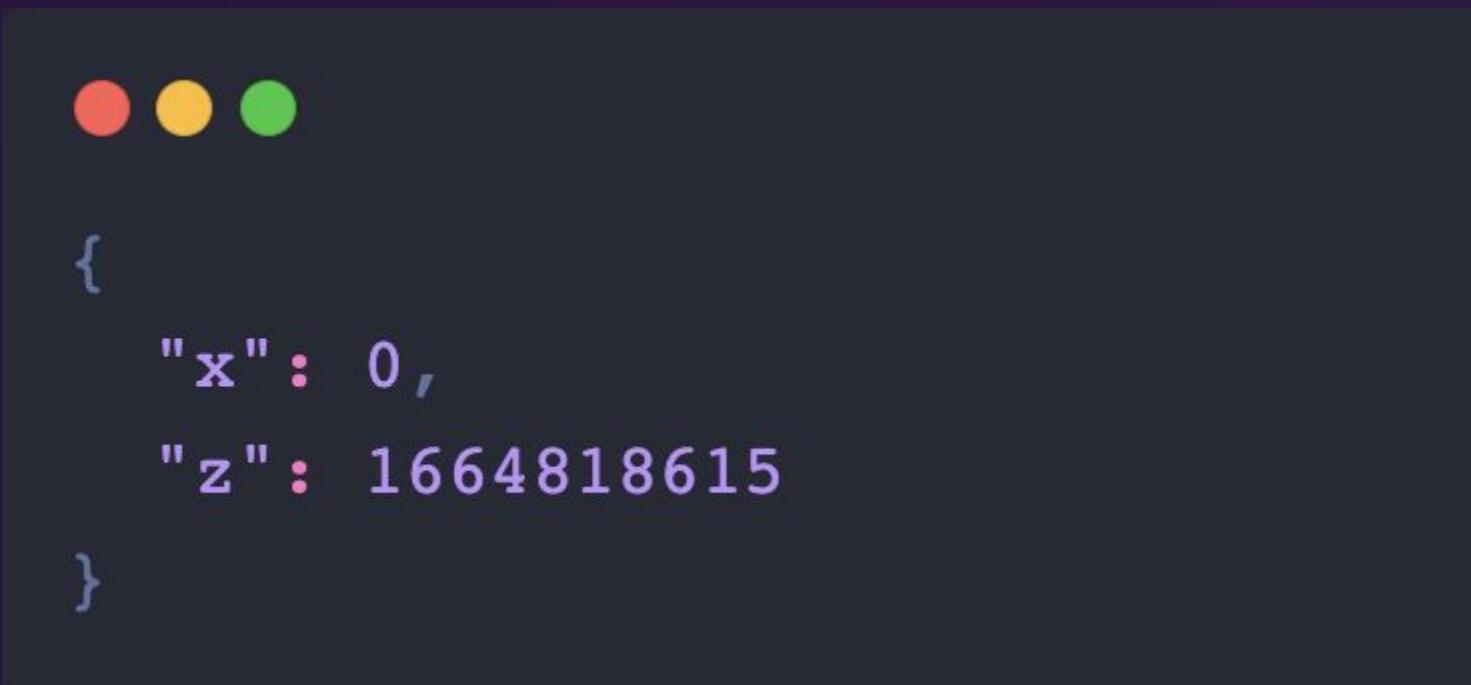
Capture enough information so you can replay the event.

STEP TWO

Generating Payload Schemas

STEP TWO

Generating Payload Schemas

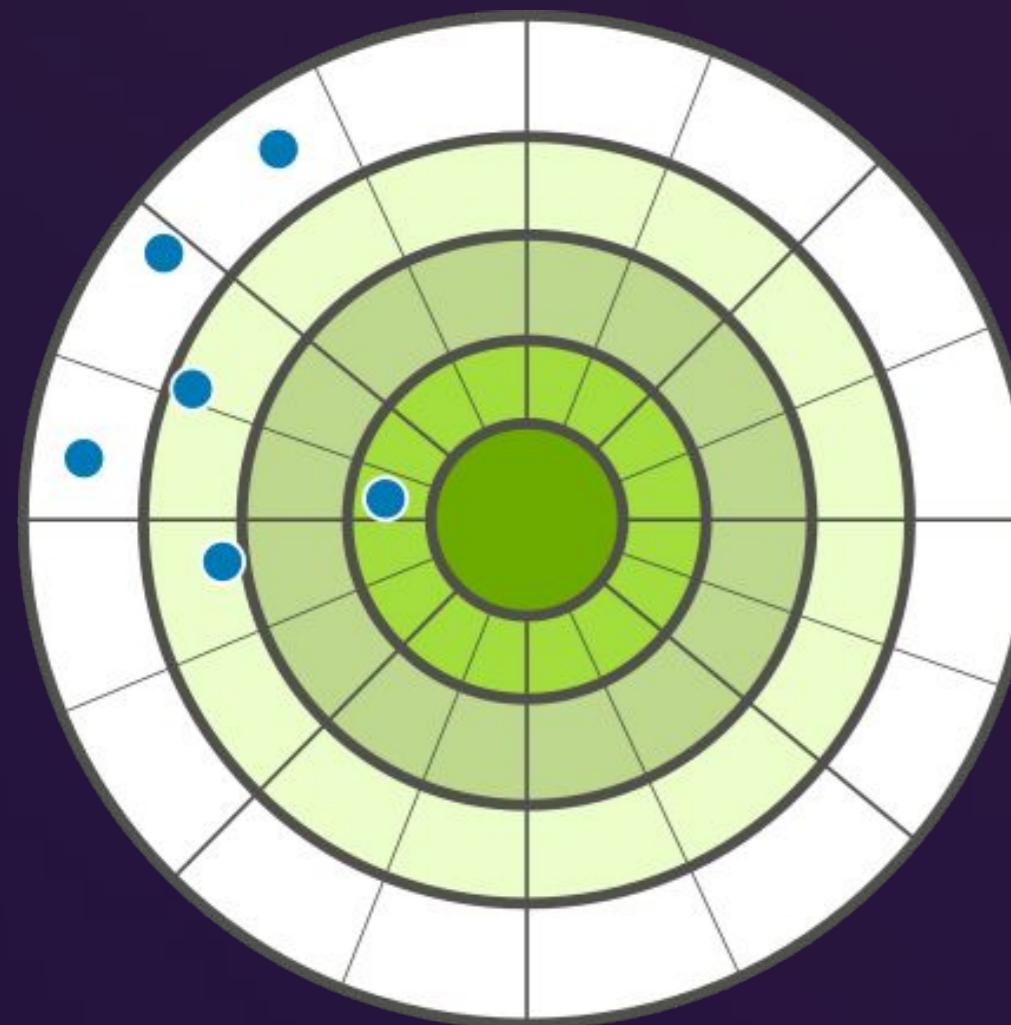


```
{
  "x": 0,
  "z": 1664818615
}
```



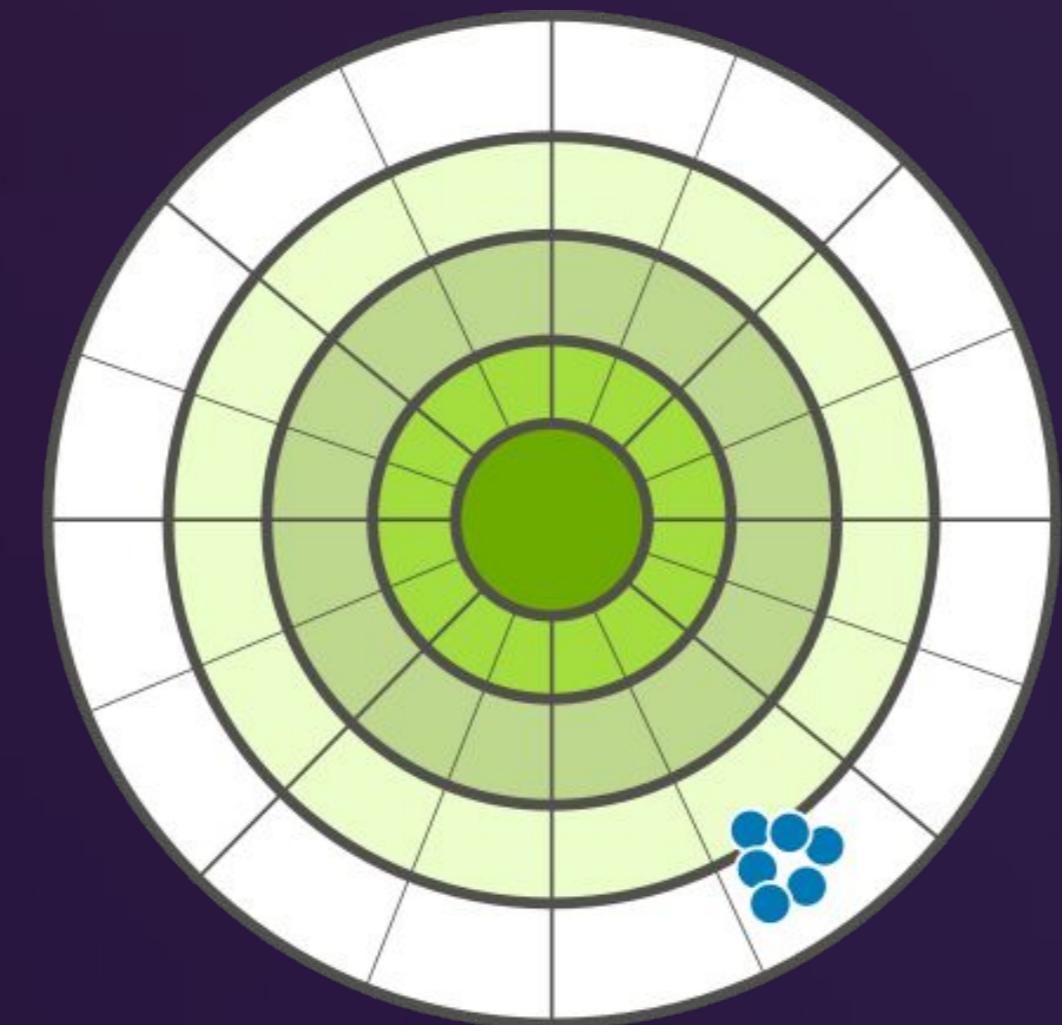
```
{
  "x": {
    "type": "integer"
  },
  "z": {
    "type": "integer"
  }
}
```

Precision vs. Accuracy



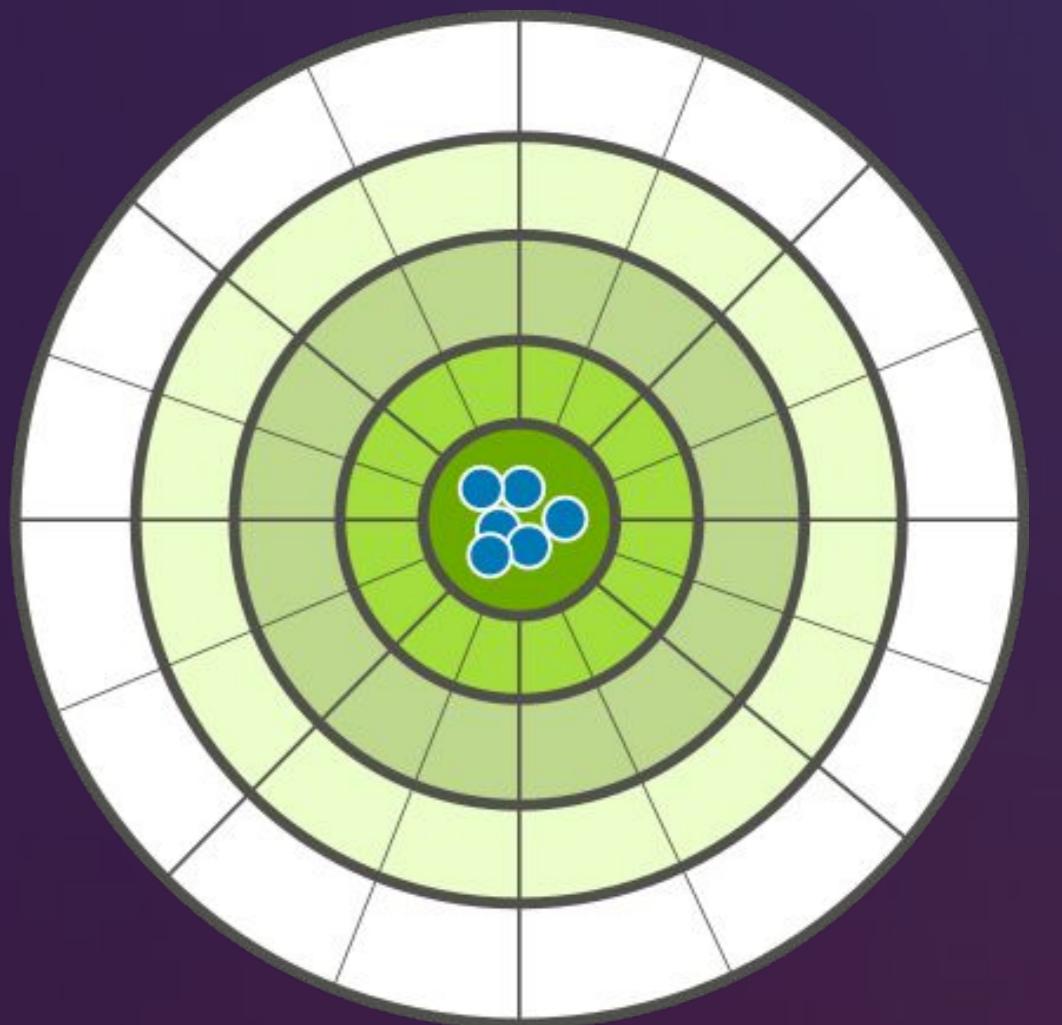
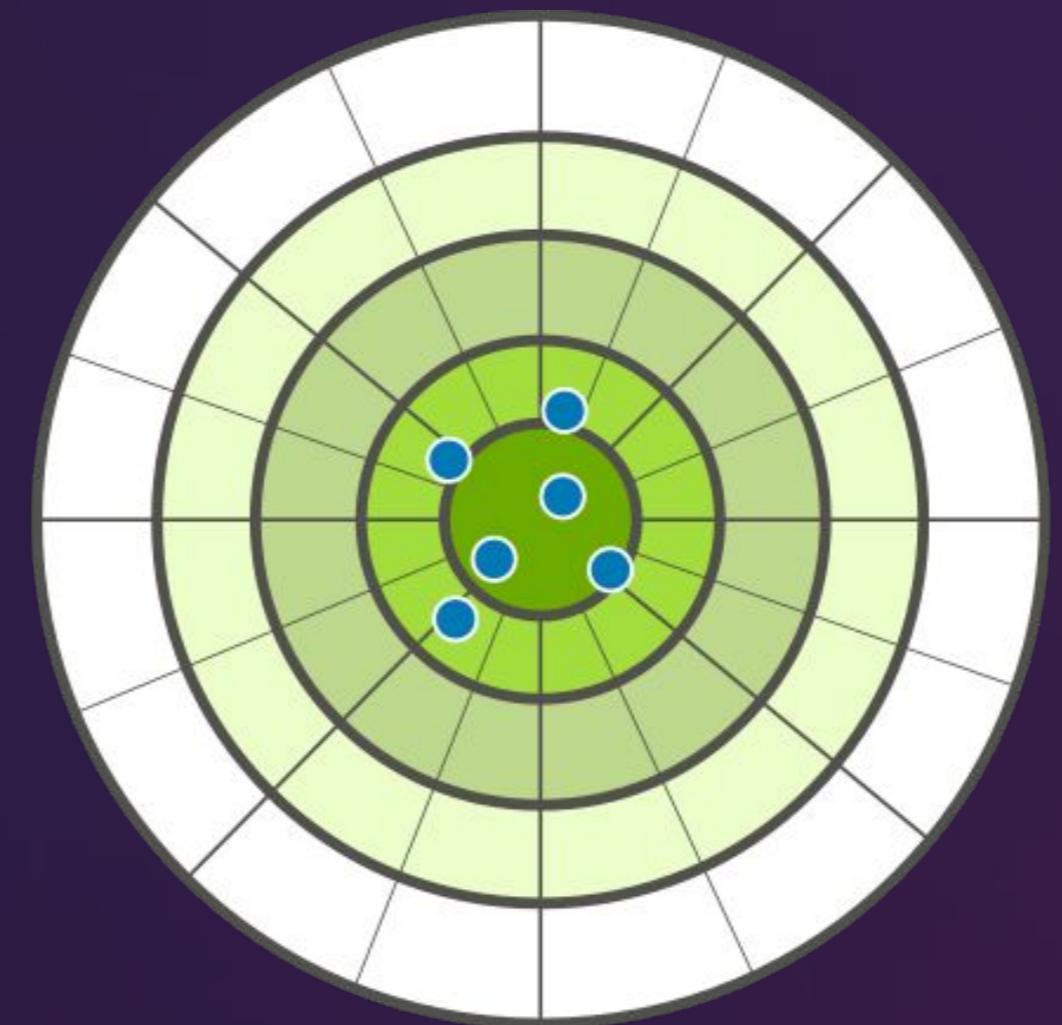
{type: string}

{minLength: 10,
maxLength: 11}



{format: date}

{???



STEP TWO - GENERATING SCHEMA PAYLOADS

High Precision, Low Accuracy

```
●●●  
  
{  
  "x": 0,  
  "z": 1664818615  
}
```

```
●●●  
  
// moxpopuli schemagen -p=_ -pa='{"x":0,"z":1664818615}'  
{  
  "properties": {  
    "x": {  
      "format": "int32",  
      "type": "integer",  
      "x-seenMaximum": 0,  
      "x-seenMinimum": 0  
    },  
    "z": {  
      "format": "timestamp",  
      "type": "integer",  
      "x-seenMaximum": 1664818615,  
      "x-seenMinimum": 1664818615  
    }  
  },  
  "type": "object",  
  "x-samples": 1  
}
```

STEP TWO - GENERATING SCHEMA PAYLOADS

Increasing Accuracy

Commutative: changing the order of operands does not change the result.

Schema is accurate for all data every processed.

```
// moxpopuli schemagen -p=file://./testdata/simplepayloads.jsonl
{
  "properties": {
    "x": {
      "format": "float",
      "type": "number",
      "x-samples": 4,
      "x-seenMaximum": 1243340323232320.5,
      "x-seenMinimum": 0
    },
    "z": {
      "oneOf": [
        {
          "format": "timestamp",
          "type": "integer",
          "x-samples": 3,
          "x-seenMaximum": 1664818617,
          "x-seenMinimum": 1664818615
        },
        {
          "type": "string",
          "x-samples": 1,
          "x-seenMaxLength": 1,
          "x-seenMinLength": 1
        }
      ]
    }
  }
}
```

STEP TWO - GENERATING SCHEMA PAYLOADS

Regaining Precision

```
// moxpopuli schemagen -p=_ -pa='{"x": "abc123"}'  
{  
  "properties": {  
    "x": {  
      "type": "string",  
      "x-seenMaxLength": 6,  
      "x-seenMinLength": 6,  
      "x-seenStrings": ["abc123"]  
    }  
  }, "type": "object", "x-samples": 1}
```

```
// moxpopuli schemagen -p=file://./testdata/ids.jsonl  
{  
  "properties": {  
    "x": {  
      "type": "string",  
      "x-identifier": true,  
      "x-samples": 6,  
      "x-seenMaxLength": 28,  
      "x-seenMinLength": 28,  
      "x-sensitive": true  
    }  
  }, "type": "object", "x-samples": 6}
```

STEP TWO - GENERATING SCHEMA PAYLOADS

Regaining Precision



```
// moxpopuli schemagen -p=file://./testdata/enums.jsonl
{
  "properties": {
    "x": {
      "enum": [
        "VALUE1",
        "VALUE2",
        "VALUE3",
        "VALUE4"
      ],
      "type": "string",
      "x-samples": 34,
      "x-seenMaxLength": 6,
      "x-seenMinLength": 6
    }
  },
  "type": "object", "x-samples": 34}
```

STEP TWO - GENERATING SCHEMA PAYLOADS

Sensitive Information



```
// moxpopuli schemagen -p=_ -pa='{"x":"94flkfw03qpf89"}'  
{  
  "properties": {  
    "x": {  
      "type": "string",  
      "x-seenMaxLength": 14,  
      "x-seenMinLength": 14,  
      "x-seenStrings": [  
        "L1LWDYU9TqVohj"  
      ],  
      "x-sensitive": true  
    }  
  }, "type": "object", "x-samples": 1}
```

STEP TWO - GENERATING SCHEMA PAYLOADS

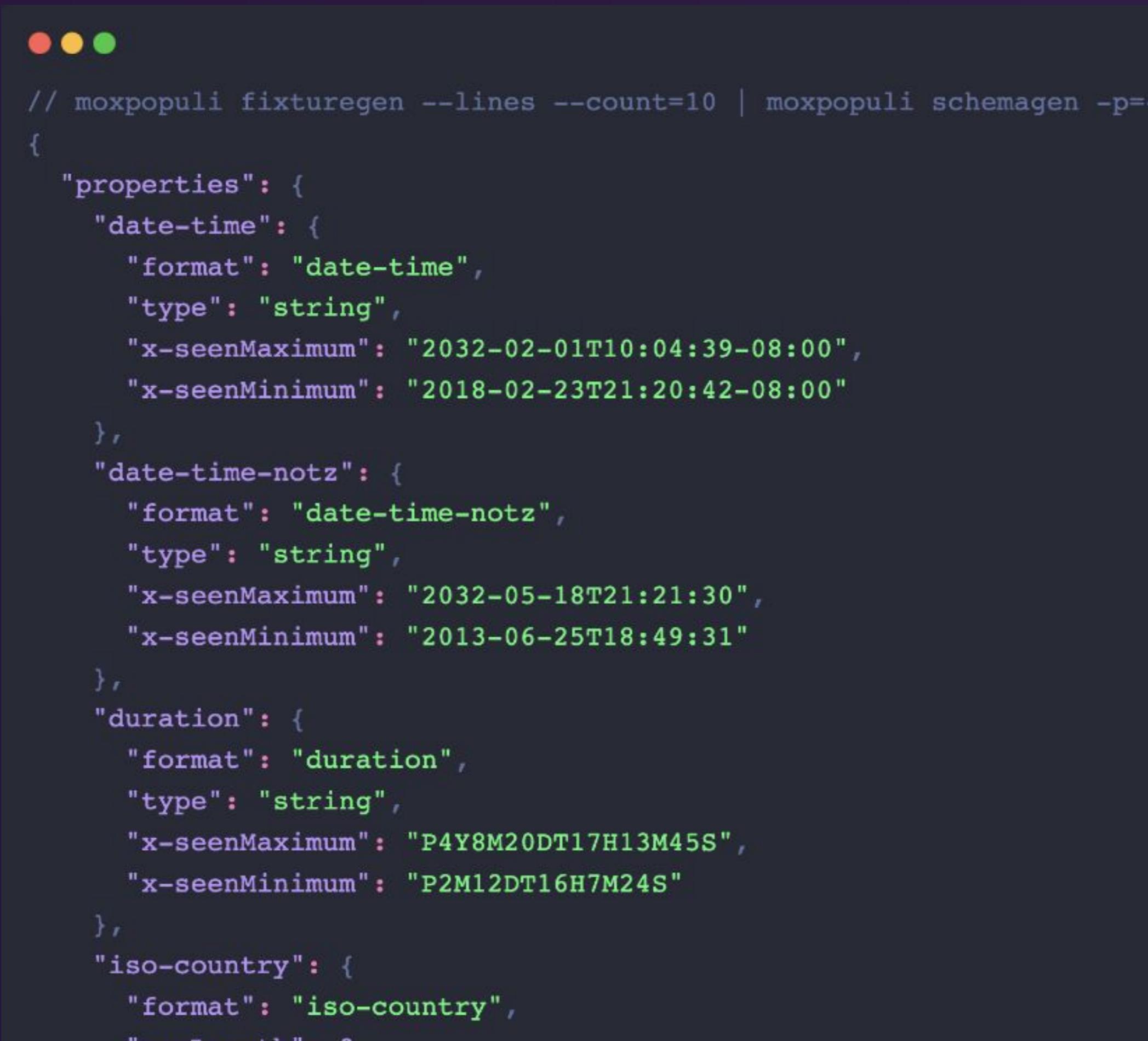
Recording Examples



```
// printf '{"id":"3bfa"}\n{"id":5}' | moxpopuli schemagen -p=- --examples=1
{
  "examples": [
    {"id": "3bfa"},
    {"id": 5}],
  "properties": {
    "id": {
      "oneOf": [
        {
          "type": "string",
          "x-seenMaxLength": 4,
          "x-seenMinLength": 4,
          "x-seenStrings": ["3bfa"]
        },
        {
          "type": "number"
        }
      ]
    }
  }
}
```

STEP TWO - GENERATING SCHEMA PAYLOADS

Accurate and Precise



```
// moxpopuli fixturegen --lines --count=10 | moxpopuli schemagen -p=-
{
  "properties": {
    "date-time": {
      "format": "date-time",
      "type": "string",
      "x-seenMaximum": "2032-02-01T10:04:39-08:00",
      "x-seenMinimum": "2018-02-23T21:20:42-08:00"
    },
    "date-time-notz": {
      "format": "date-time-notz",
      "type": "string",
      "x-seenMaximum": "2032-05-18T21:21:30",
      "x-seenMinimum": "2013-06-25T18:49:31"
    },
    "duration": {
      "format": "duration",
      "type": "string",
      "x-seenMaximum": "P4Y8M20DT17H13M45S",
      "x-seenMinimum": "P2M12DT16H7M24S"
    },
    "iso-country": {
      "format": "iso-country",
      "x-seenMaximum": "AD"
    }
  }
}
```

```
"numerical": {
  "format": "numerical",
  "type": "string",
  "x-seenMaximum": "762129038907",
  "x-seenMinimum": "-935736330713"
},
"timestamp": {
  "format": "timestamp",
  "type": "integer",
  "x-seenMaximum": 1925840647,
  "x-seenMinimum": 110039324
},
"timestamp-ms": {
  "format": "timestamp-ms",
  "type": "integer",
  "x-seenMaximum": 1847512946466,
  "x-seenMinimum": 95515572188
},
"uuid4": {
  "format": "uuid4",
  "maxLength": 36,
  "minLength": 36,
  "type": "string",
  "x-sensitive": true
},
"zero-one": {
  "enum": [0, 1],
  "format": "zero-one",
  "type": "integer"
}
```

STEP THREE

Generating an AsyncAPI Spec

STEP THREE - REST OF THE SPEC

What can we generate?

- | Servers
- | Channels and Channel Items
- | Subscribe Operations
- | Messages



**HTTP
AHEAD**

STEP THREE - REST OF THE SPEC

Anatomy of an HTTP Event

- Path \Rightarrow channels[request path]
- Method \Rightarrow channel.bindings.http.method
- Body \Rightarrow message.payload
- Headers \Rightarrow Special, Protocol, Application

STEP THREE - REST OF THE SPEC

Special Headers

- | “Host”, “Accept” \Rightarrow api.webhookdb.com, http 1.1
- | “Content-Type” \Rightarrow message.contentType
- | “Request-Id” \Rightarrow message.correlationId

STEP THREE - REST OF THE SPEC

Protocol Headers

```
● ● ●

{
  "message": {
    "bindings": {
      "http": {
        "Accept-Encoding": {
          "type": "string",
          "x-lastValue": "gzip;q=1.0,deflate;q=0.6,identity;q=0.3"
        },
        "Host": {
          "type": "string",
          "x-lastValue": "localhost:18001"
        },
        "User-Agent": {
          "type": "string",
          "x-lastValue": "WebhookDB/v1 webhookdb.com 2022-10-01T00:00:00Z"
        },
        "Version": {
          "type": "string",
          "x-lastValue": "HTTP/1.1"
        }
      }
    }
  }
}
```

STEP THREE - REST OF THE SPEC

Application Headers



```
{  
  "channels": {  
    "/v1/service_integrations/svi_81f5em7skqagk7pstse7b4j1r": {  
      "subscribe": {  
        "message": {  
          "headers": {  
            "properties": {  
              "Whdb-Secret": {  
                "type": "string",  
                "x-identifier": true,  
                "x-samples": 7,  
                "x-seenMaxLength": 25,  
                "x-seenMinLength": 25,  
                "x-sensitive": true  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

STEP THREE - REST OF THE SPEC

Example Spec

```
○ ○ ○

{
  "asyncapi": "2.4.0",
  "info": {
    "contact": {
      "email": "hello@webhookdb.com",
      "name": "Hello"
    },
    "description": "These are the WebhookDB endpoints av",
    "termsOfService": "https://webhookdb.com/terms/",
    "title": "WebhookDB Integrations for Demo Org",
    "version": "1.0.0"
  },
  "servers": {
    "localhost:18001": {
      "protocol": "http",
      "protocolVersion": "1.1",
      "url": "localhost:18001"
    }
  },
  "channels": {
    "/v1/service_integrations/svi_81f5em7skqagk7pstse7b4": {
      "subscribe": {
        "bindings": {
          "http": {
            "method": "POST",
            "type": "request"
          }
        },
        "message": {
          "bindings": {
            "http": {
              "headers": {
                "properties": {
                  "Accept": {
                    "type": "string",
                    "x-lastValue": "*/*"
                  }
                }
              },
              "type": "object"
            }
          }
        }
      }
    }
  }
}
```

STEP FOUR

Mox Populi Fixturing

STEP FOUR - FIXTURING

Generating From JSONSchema



```
// moxpopuli datagen --l=file://./testdata/fixturedemo.schema.json
{
  "SessoinIP": "30.34.254.115",
  "array-of-ids": [
    "14f35dd7-ddbd-5238-dc14-1fb7ec03b5bc",
    "b4fa9ab3-4c59-629c-9019-f87272617e90"
  ],
  "arrayofobjects": [ {"myid": "7f968714-d8af-55fd-9e44-c9e710bb7c8f"} ],
  "base64bytes": "d3c06f1859fd1a9e82d5c8",
  "currency": "XBB",
  "databaseid": "937",
  "email": "xiMWdMU@luaDPwC.com",
  "ended_at": "2024-08-08T02:00:00-07:00",
  "homepage": "https://PQdXgBt.info/xuZlXoA",
  "started_on": "2021-09-30"
}
```

STEP FOUR - FIXTURING

From an AsyncAPI Spec

```
// moxpopuli vox -l=file://./testdata/whdbspec.json --count=100 --print

REQUEST /v1/service_integrations/svi_ct14kxb4ngg3auyrysjwzjlk5-0
POST /v1/service_integrations/svi_ct14kxb4ngg3auyrysjwzjlk5 HTTP/1.1
Host: localhost:18001
User-Agent: my awesome app
Content-Length: 284
Accept: */*
Accept-Encoding: gzip
Connection: close
Content-Type: application/json
Trace-Id: d274eb7f-d4a2-e1f8-17f7-cdd7c97587f7
Version: HTTP/1.1
Whdb-Secret: c13cc9c79c6fb8347c7aba14

{
  "created_at": "2022-09-14T20:14:39-07:00",
  "email": "oGQtORV@YsWSHXP.ru",
  "id": 20,
  "name": "55e3a8408bc5f8d6e961",
  "note": "",
  "opaque_id": "55e3a8408bc5f8d6e961"
}

RESPONSE /v1/service_integrations/svi_ct14kxb4ngg3auyrysjwzjlk5-0
HTTP/1.0 200 OK
Connection: close
Content-Type: application/json
Vary: Origin

{
  "o": "k"
}
```

CONCLUSION

Takeaways



AsyncAPI can describe any
kind of API.



**Don't build it yourself if you
want to be lazy.**

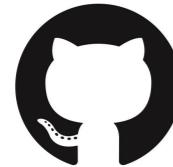
A photograph of four children playing soccer in a dense, misty forest. The children are in various stages of action: one in a green shirt is jumping to head the ball, another in a blue shirt is running towards the ball, a third in a red and blue shirt is also jumping, and a fourth in a blue shirt is crouching low to the ground. The ball is positioned above the jumping child in the green shirt. The forest is filled with tall, thin trees and a thick mist, creating a serene and slightly mysterious atmosphere.

It's most important to have fun.

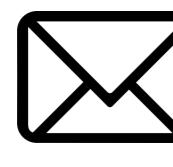
WEBHOOKDB & MOX POPULI

Thank You!

 www.webhookdb.com

 github.com/lithictech/moxpopuli

 twitter.com/robgalanakis

 rob@webhookdb.com
natalie@webhookdb.com

 Portland, Oregon, USA

