



**Ricardo Signes (rjbs)**  
**FOSDEM 2026 Presentation**  
**Presented in TohoScope**

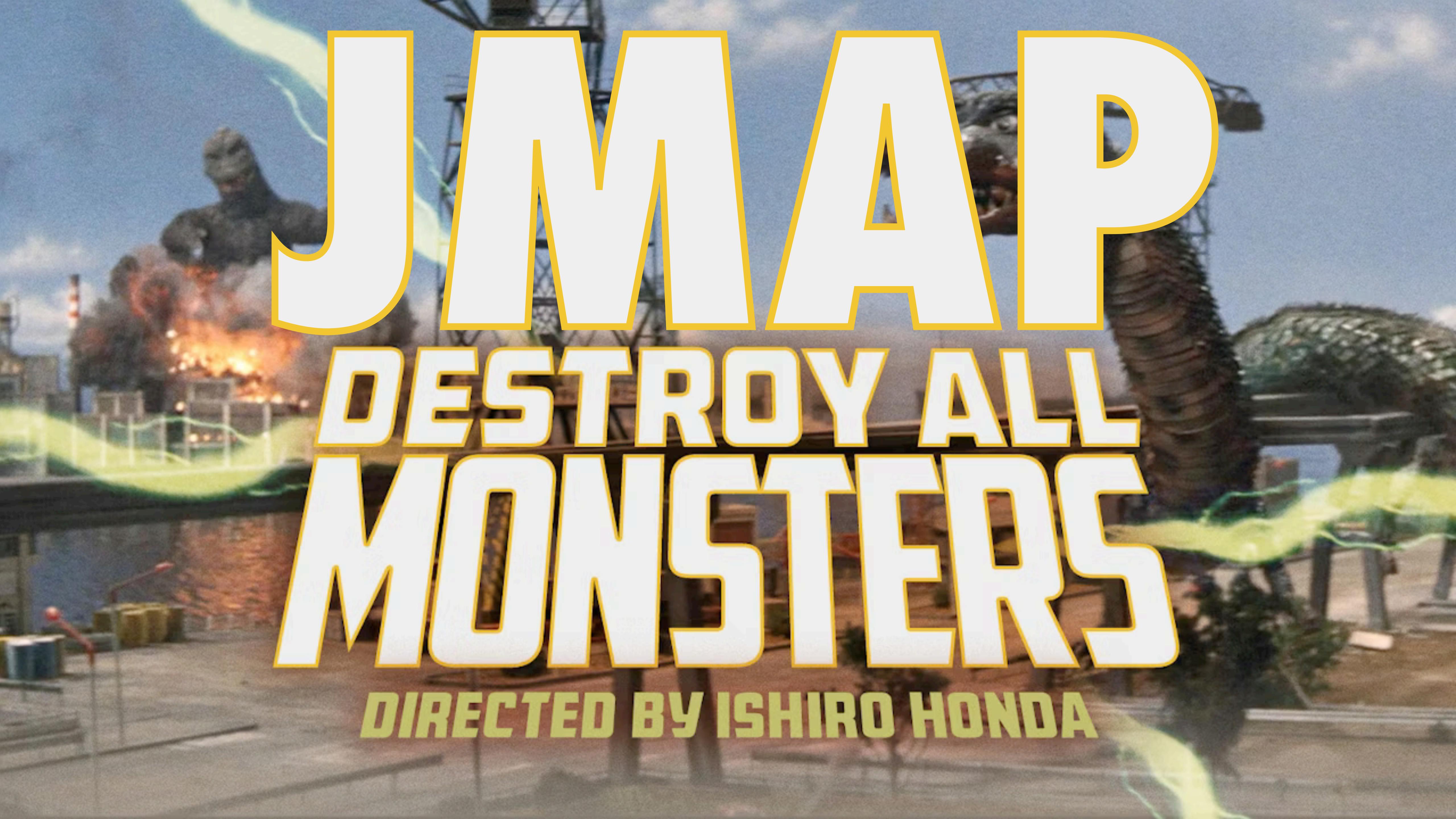


# **DESTROY ALL MONSTERS**

**DIRECTED BY ISHIRO HONDA**

# JUMAIP DESTROY ALL MONSTERS

DIRECTED BY ISHIRO HONDA





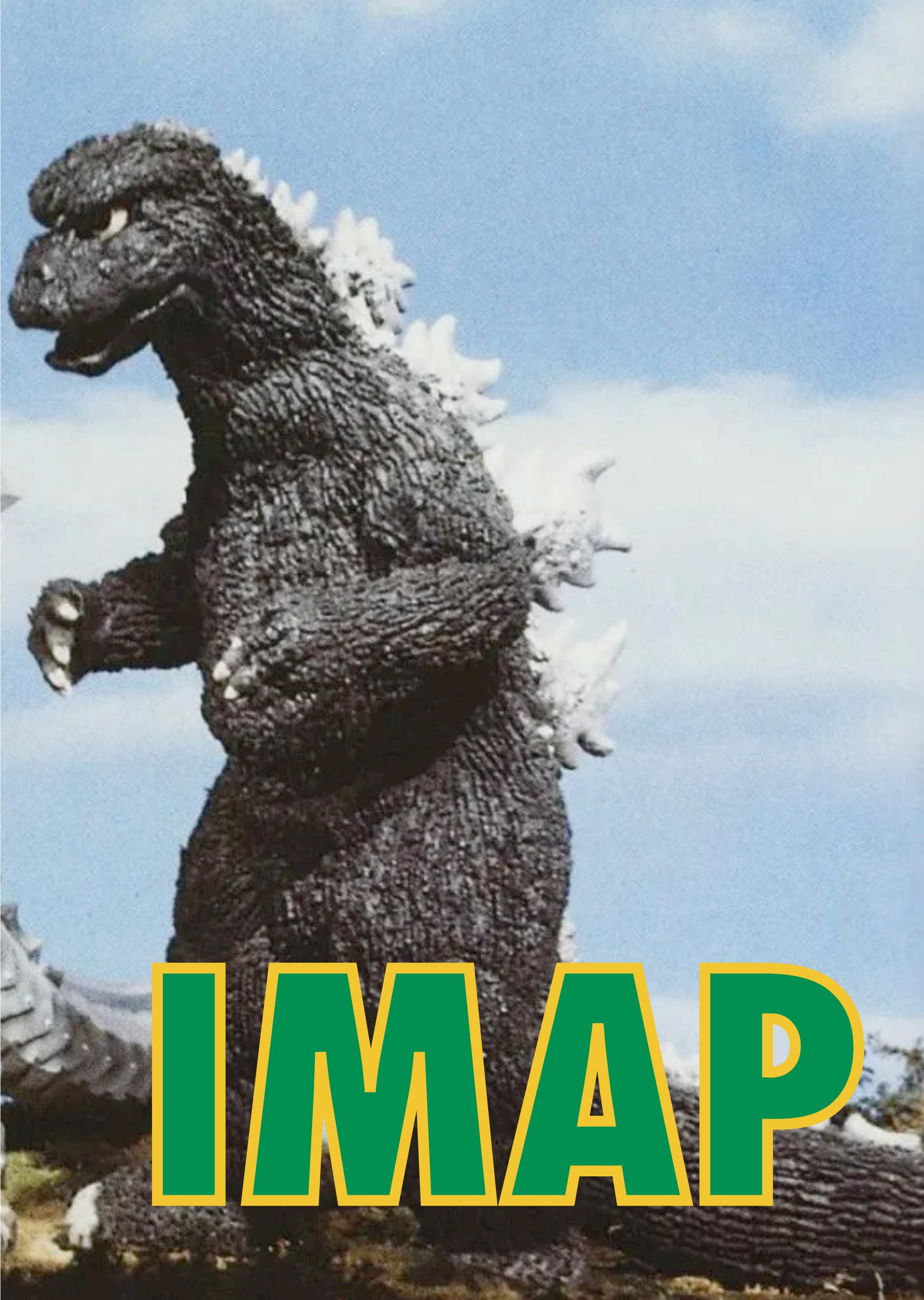
# TERROR OF IMAP

aka "JMAP the absolute minimum.pdf"

fosdem, 2024

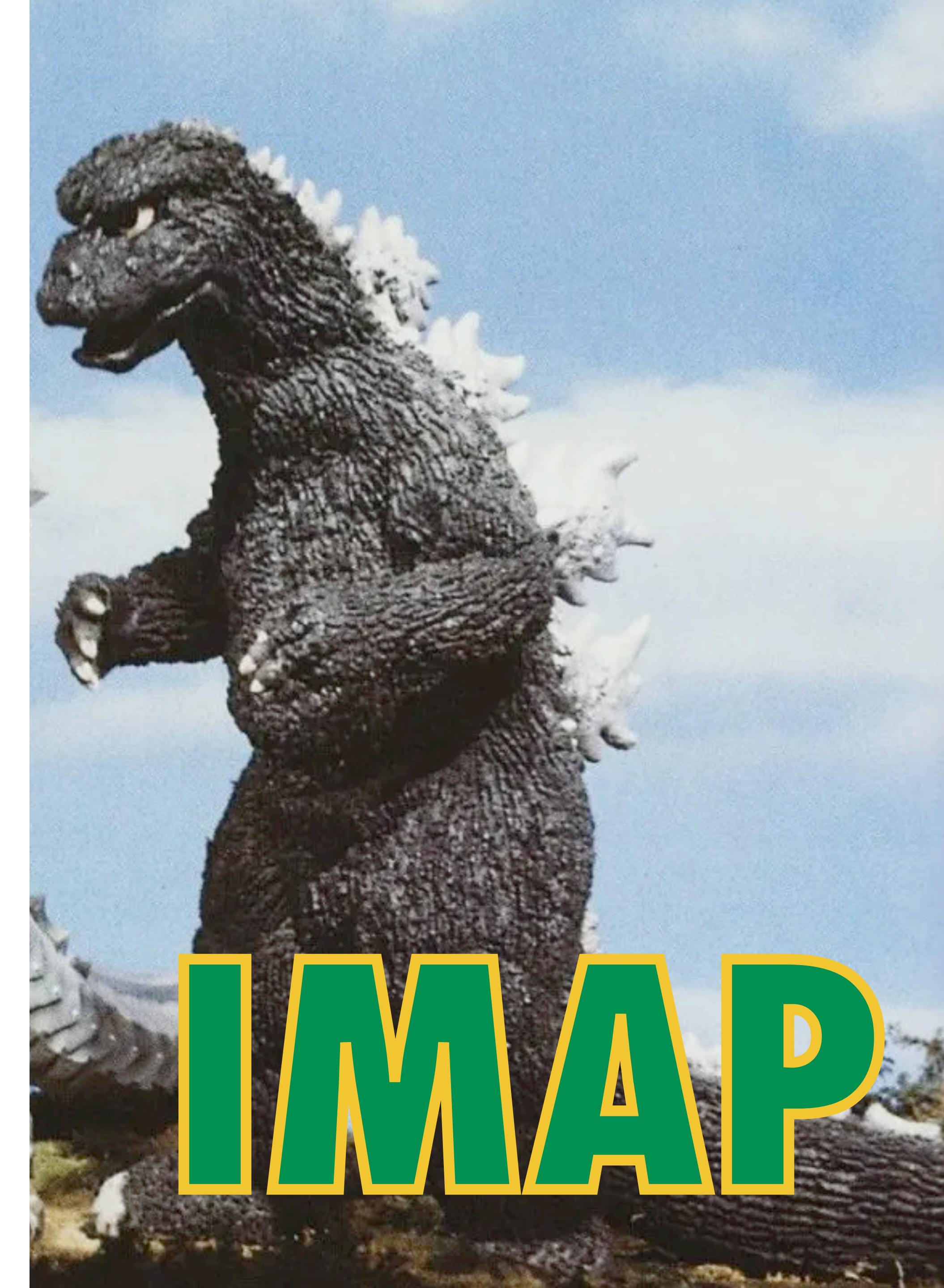


**IMAP**

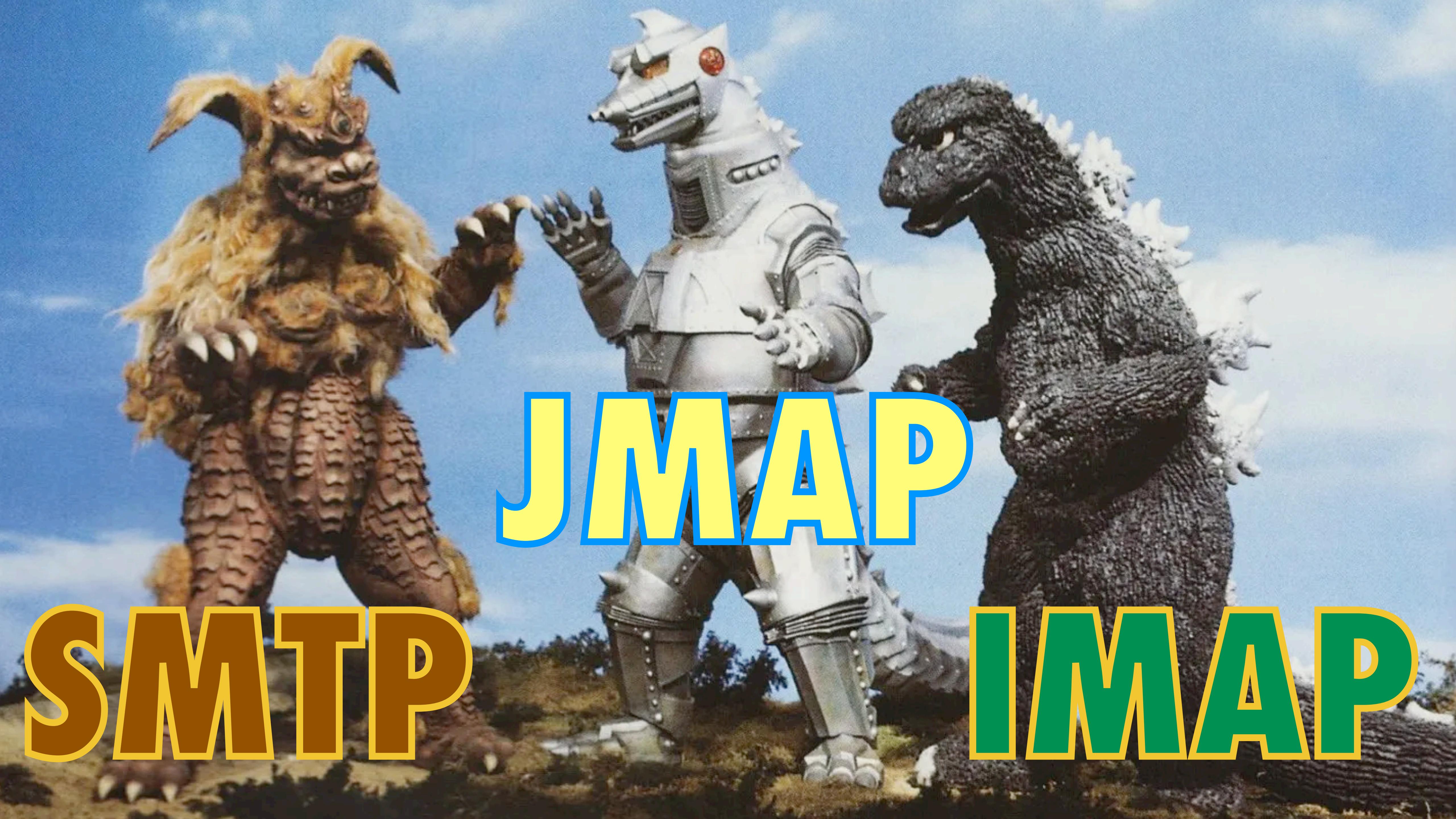




**SMTP**



**IMAP**



**SMTP**

**JMAP**

**IMAP**

if you're not a kaiju movie fan,  
you might be confused right now

**Godzilla and King Caesar**  
are horrible monsters that lie  
beneath the modern world,  
and while they serve a purpose, they  
are also *the worst*.

## IMAP and SMTP

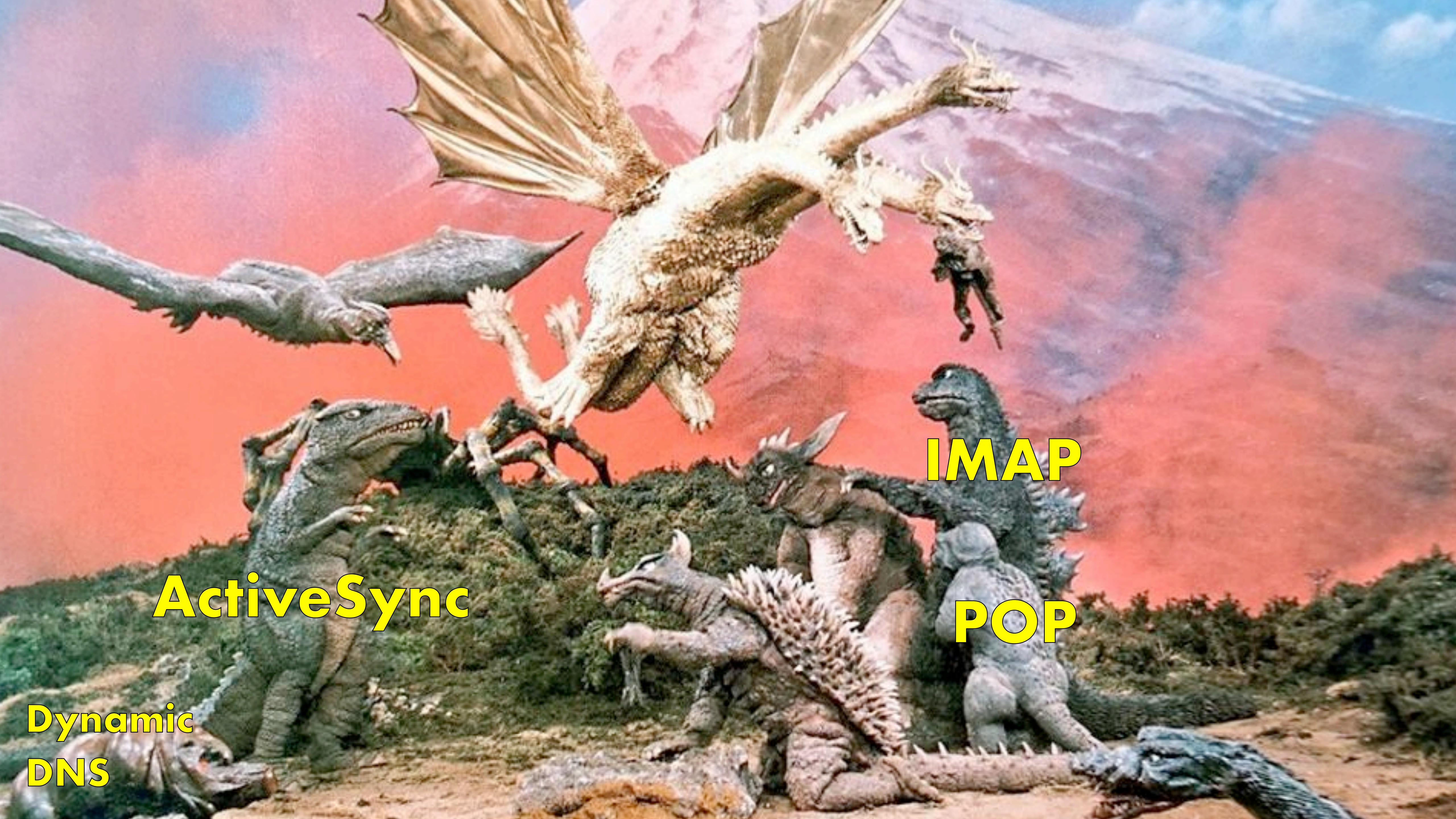
are horrible monsters that lie beneath the modern world, and while they serve a purpose, they are also *the worst*.

Mechagodzilla was sent to earth  
from Space to  
save us from the monsters.

JMAP was sent to earth  
from Fastmail to  
save us from the protocols.

those aren't the only two  
monsters, though





**ActiveSync**

**Dynamic  
DNS**

**IMAP**

**POP**



THIS  
GUY

ActiveSync

Dynamic  
DNS

IMAP  
POP



All rights reserved. © 2013 Legendary Pictures Entertainment All rights reserved. LEGENDARY



GHIDORAH

All rights reserved. LEGENDARY



# KING GHIDORAH

All rights reserved. © 2019 Legendary Pictures Entertainment, LLC. All rights reserved. LEGENDARY

**but first, IMAP**

connection established

S \* OK IMAP4rev1 Service Ready

C a001 login mrc secret

S a001 OK LOGIN completed

C a002 select inbox

S \* 18 EXISTS

S \* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)

S \* 2 RECENT

S \* OK [UNSEEN 17] Message 17 is the first unseen message

S \* OK [UIDVALIDITY 3857529045] UIDs valid

S a002 OK [READ-WRITE] SELECT completed

unsolicited  
command tag

response

list

response code

```
C A03 SELECT INBOX (QRESYNC (67890007 123 41,43:211,214:541))  
S * OK [CLOSED]  
S * 314 EXISTS  
S * 15 RECENT  
S * OK [UIDVALIDITY 67890007] UIDVALIDITY  
S * OK [UIDNEXT 567] Predicted next UID  
S * OK [HIGHESTMODSEQ 130] ← server state  
S * OK [UNSEEN 7] There are some unseen messages in the mailbox  
S * FLAGS (\Answered \Flagged \Draft \Deleted \Seen)  
S * OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]  
S * VANISHED (EARLIER) 41,43:116,118,120:211,214:540  
S * 49 FETCH (UID 117 FLAGS (\Seen \Answered) MODSEQ (128))  
S * 50 FETCH (UID 119 FLAGS (\Draft $MDNSent) MODSEQ (129))  
S ...  
S * 100 FETCH (UID 541 FLAGS (\Seen $Forwarded) MODSEQ (130))  
S A03 OK [READ-WRITE] mailbox selected
```

resync

client state

object state

---

```
C Give me updates to inbox since 123.  
S Here are many updates. Apply these  
and you will be at 130.
```

Right!

The good stuff is good,  
but the bad stuff is  
*a total buzzkill.*



# JMAP

JSON Meta Application Protocol

My goal is not to teach you how  
to use JMAP today.

My goal is not to teach you how  
to use JMAP today.

I only have 25 minutes!

My goal is not to teach you how  
to use JMAP today.

I only have 25 minutes!

My goal is to show you it's easy  
and worthwhile to learn.

request

```
POST /jmap HTTP/1.1
Host: api.fastmail.com
Content-Type: application/json

{
    ...
    "Email/get",
    { "ids": [ "1", "2", "3" ] },
    ...
}
```

response

Content-Type: application/json

```
{ ...
  ...
  "Email/get",
  {
    ...
    "list": [
      {
        ...
        "id": "1",
        "subject": "Hi, Brussels! 🇧🇪❤️",
      }
    ],
  },
  ...
}
```

request

```
{  ...
  "Email/changes",
  {
    ...
      "sinceState": "616"
  },
  ...
}
```

request

```
{ ...
  "Email/changes",
  {
    ...
      "sinceState": "616"
  },
  ...
}
```

response

```
{ ...
  "Email/changes",
  {
    ...
      "oldState": "616",
      "newState": "717",
      "created": [ "a", "b" ],
      "updated": [ "c" ],
      "destroyed": []
  },
  ...
}
```

JMAP is easy.

JSON + HTTP = all your data

offline sync is not trivial, but easy

now that you understand JMAP,  
back to monster zero



All rights reserved. © 2013 Legendary Pictures Entertainment All rights reserved. LEGENDARY



GHIDORAH

All rights reserved. LEGENDARY



# KING GHIDORAH

All rights reserved. © 2019 Legendary Pictures Entertainment, LLC. All rights reserved. LEGENDARY





CardDAV



**CardDAV**

**CalDAV**



**CardDAV** **davDAV** **CalDAV**



PROPFIND /dav/calendars/user/rjbs/ HTTP/1.0

Host: caldav.fastmail.com

Brief: t

Authorization: Basic ...

Accept-Language: en-us

Accept-Encoding: gzip, deflate, br

Content-Type: text/xml

Depth: 1

User-Agent: YourCoolCalendar/1.2

Prefer: return=minimal

**PROPFIND /dav/calendars/user/rjbs/ HTTP/1.0**

Host: caldav.fastmail.com

Brief: t

Authorization: Basic ...

Accept-Language: en-us

Accept-Encoding: gzip, deflate, br

Content-Type: text/xml

Depth: 1

User-Agent: YourCoolCalendar/1.2

Prefer: return=minimal

HTTP/1.0 207 Multi-Status

Date: Wed, 23 Sep 2020 01:33:12 GMT

Connection: close

Vary: Accept-Encoding, Brief, Prefer

Preference-Applied: return=minimal

Content-Type: application/xml; charset=utf-8

HTTP/1.0 207 Multi-Status

Date: Wed, 23 Sep 2020 01:33:12 GMT

Connection: close

Vary: Accept-Encoding, Brief, Prefer

Preference-Applied: return=minimal

Content-Type: application/xml; charset=utf-8

# HTTP/1.0 207 Multi-Status

...

```
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/col/abc/</D:href>
    <D:propstat>
      ...
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>http://www.example.com/col/xyz/</D:href>
    <D:propstat>
      ...
      <D:status>HTTP/1.1 418 I AM A TEAPOT</D:status>
    </D:propstat>
  </D:response>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<A:propfind xmlns:A="DAV:">
  <A:prop>
    <A:add-member/>
    <C:allowed-sharing-modes xmlns:C="http://calendarsERVER.org/ns/" />
    <D:autoprovisioned xmlns:D="http://apple.com/ns/ical/" />
    <E:bulk-requests xmlns:E="http://me.com/_namespace/" />
    <B:calendar-alarm xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <D:calendar-color xmlns:D="http://apple.com/ns/ical/" />
    <B:calendar-description xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <B:calendar-free-busy-set xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <D:calendar-order xmlns:D="http://apple.com/ns/ical/" />
    <B:calendar-timezone xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <A:current-user-privilege-set/>
    <B:default-alarm-vevent-date xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <B:default-alarm-vevent-datetime xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <A:displayname/>
    <C:getctag xmlns:C="http://calendarsERVER.org/ns/" />
  ...

```

```
<?xml version="1.0" encoding="UTF-8"?>
<A:propfind xmlns:A="DAV:">
  <A:prop>
    <A:add-member/>
    <C:allowed-sharing-modes xmlns:C="http://calendarsERVER.org/ns/" />
    <D:autoprovisioned xmlns:D="http://apple.com/ns/ical/" />
    <E:bulk-requests xmlns:E="http://me.com/_namespace/" />
    <B:calendar-alarm xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <D:calendar-color xmlns:D="http://apple.com/ns/ical/" />
    <B:calendar-description xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <B:calendar-free-busy-set xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <D:calendar-order xmlns:D="http://apple.com/ns/ical/" />
    <B:calendar-timezone xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <A:current-user-privilege-set/>
    <B:default-alarm-vevent-date xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <B:default-alarm-vevent-datetime xmlns:B="urn:ietf:params:xml:ns:caldav" />
    <A:displayname/>
    <C:getctag xmlns:C="http://calendarsERVER.org/ns/" />
  ...

```

...and if you don't mind XML...

```
<A:privilege>
  <CY:admin/>
</A:privilege>
</A:current-user-privilege-set>
<B:default-alarm-vevent-date><! [CDATA[BEGIN:VALARM
X-WR-ALARMUID:4C2905A7-3539-49D8-B7BA-A6220A7913C7
UID:4C2905A7-3539-49D8-B7BA-A6220A7913C7
TRIGGER:-PT15H
ATTACH;VALUE=URI:Basso
ACTION:AUDIO
END:VALARM
]]></B:default-alarm-vevent-date>
<B:default-alarm-vevent-datetime><! [CDATA[BEGIN:VALARM
X-WR-ALARMUID:682FAB1B-335F-4443-A75E-BA704074F436
UID:682FAB1B-335F-4443-A75E-BA704074F436
TRIGGER;VALUE=DATE-TIME:19760401T005545Z
ACTION:NONE
END:VALARM
]]></B:default-alarm-vevent-datetime>
<A:displayname><! [CDATA[Ricardo Signes]]></A:displayname>
<C:invite/>
```

```
    <A:privilege>
        <CY:admin/>
    </A:privilege>
</A:current-user-privilege-set>
<B:default-alarm-vevent-date><! [CDATA[BEGIN:VALARM
X-WR-ALARMUID:4C2905A7-3539-49D8-B7BA-A6220A7913C7
UID:4C2905A7-3539-49D8-B7BA-A6220A7913C7
TRIGGER:-PT15H
ATTACH;VALUE=URI:Basso
ACTION:AUDIO
END:VALARM
]]></B:default-alarm-vevent-date>
<B:default-alarm-vevent-datetime><! [CDATA[BEGIN:VALARM
X-WR-ALARMUID:682FAB1B-335F-4443-A75E-BA704074F436
UID:682FAB1B-335F-4443-A75E-BA704074F436
TRIGGER;VALUE=DATE-TIME:19760401T005545Z
ACTION:NONE
END:VALARM
]]></B:default-alarm-vevent-datetime>
<A:displayname><! [CDATA[Ricardo Signes]]></A:displayname>
<C:invite/>
```

```
<A:privilege>
  <CY:admin/>
</A:privilege>
</A:current-user-privilege-set>
<B:default-alarm-vevent-date><! [CDATA[BEGIN:VALARM
X-WR-ALARMUID:4C2905A7-3539-49D8-B7BA-A6220A7913C7
UID:4C2905A7-3539-49D8-B7BA-A6220A7913C7
TRIGGER:-PT15H
ATTACH;VALUE=URI:Basso
ACTION:AUDIO
END:VALARM
]]></B:default-alarm-vevent-date>
<B:default-alarm-vevent-datetime><! [CDATA[BEGIN:VALARM
X-WR-ALARMUID:682FAB1B-335F-4443-A75E-BA704074F436
UID:682FAB1B-335F-4443-A75E-BA704074F436
TRIGGER;VALUE=DATE-TIME:19760401T005545Z
ACTION:NONE
END:VALARM
]]></B:default-alarm-vevent-datetime>
<A:displayname><! [CDATA[Ricardo Signes]]></A:displayname>
<C:invite/>
```

## iCalendar: the prize inside the XML

```
BEGIN:VALARM  
X-WR-ALARMUID:682FAB1B-335F-4443-A75E-BA704074F436  
UID:682FAB1B-335F-4443-A75E-BA704074F436  
TRIGGER;VALUE=DATE-TIME:19760401T005545Z  
ACTION:NONE  
END:VALARM
```

## iCalendar: the prize inside the XML

```
BEGIN:VALARM
X-WR-ALARMUID:682FAB1B-335F-4443-A75E-BA704074F436
UID:682FAB1B-335F-4443-A75E-BA704074F436
TRIGGER;VALUE=DATE-TIME:19760401T005545Z
ACTION:NONE
END:VALARM
```

## iCalendar: the prize inside the XML

```
BEGIN:VALARM
  X-WR-ALARMUID: 682FAB1B-335F-4443-A75E-BA704074F436
  UID:           682FAB1B-335F-4443-A75E-BA704074F436
  TRIGGER;VALUE=DATE-TIME : 19760401T005545Z
  ACTION:        NONE
END:VALARM
```

## iCalendar: the prize inside the XML

BEGIN:VALARM

X-WR-ALARMUID: 682FAB1B-335F-4443-A75E-BA704074F436

UID: 682FAB1B-335F-4443-A75E-BA704074F436

TRIGGER VALUE=DATE-TIME : 19760401T005545Z

ACTION: NONE

END:VALARM

# Component

BEGIN:VALARM  
X-WR-ALARMUID:  
UID:  
TRIGGER  
ACTION:  
END:VALARM

Property name

Property value

Parameter name

Parameter value

VALUE=DATE-TIME : 19760401T005545Z

NONE

**So, CardDAV is...**

iCalendar

iCalendar

embedded in

XML

iCalendar

embedded in

XML

synced over

DAV

iCalendar

embedded in

XML

synced over

DAV

which is deformed

HTTP

iCalendar/vCard

embedded in

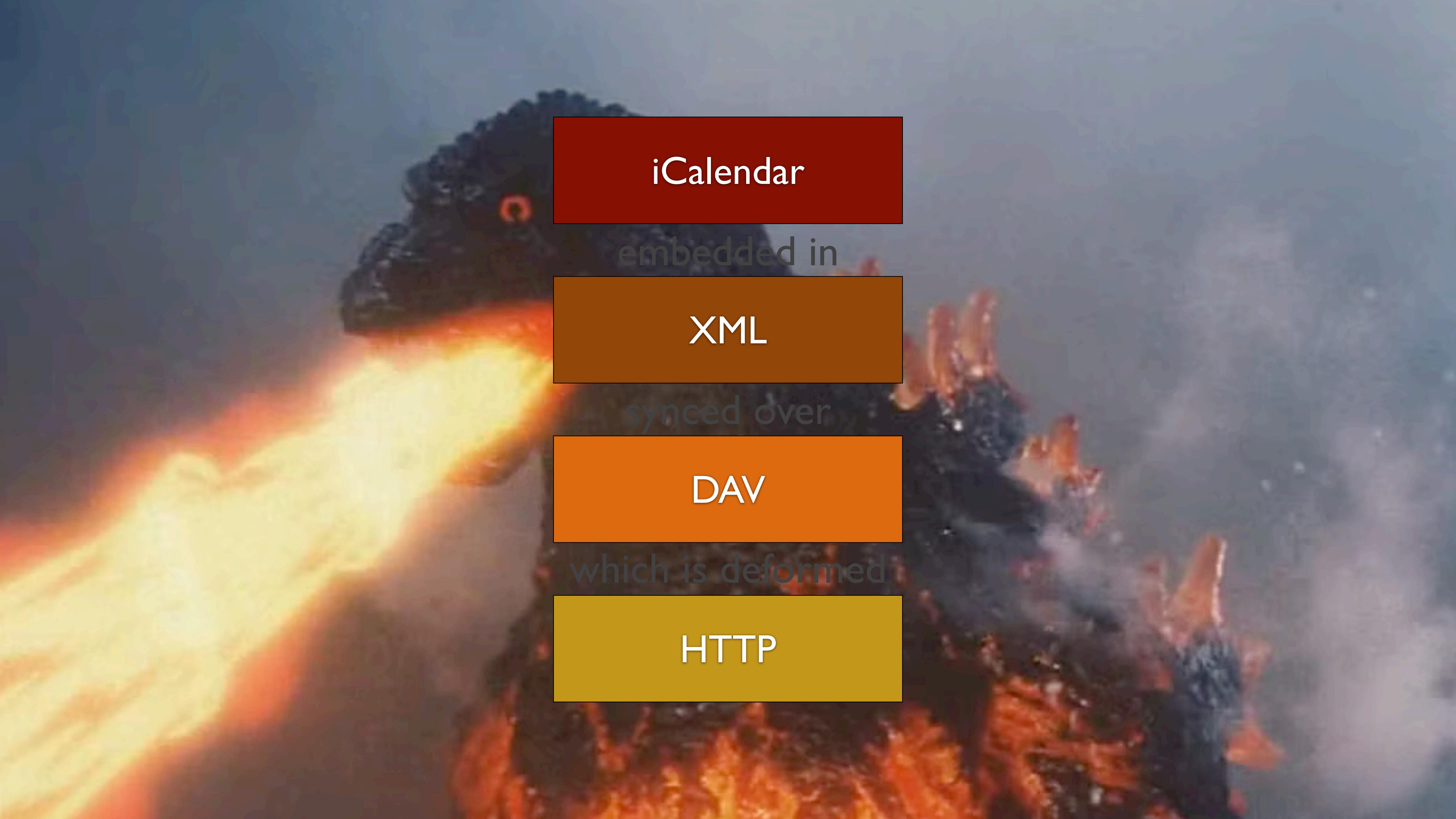
XML

synced over

DAV

which is deformed

HTTP

The background of the slide features a dramatic scene of a large, dark, smoky monster, possibly Godzilla, with a single red eye and sharp claws, set against a background of fire and smoke.

iCalendar

embedded in

XML

synced over

DAV

which is deformed

HTTP

```
<DAV:sync-token>
  data: ,1571793414-7647006
</DAV:sync-token>
```

REPORT /dav/calendars/user/rjbs/my-cool-calendar/ HTTP/1.0  
Host: caldav.fastmail.com  
Connection: close  
Content-Length: 215  
Content-Type: text/xml  
Depth: 1  
Accept-Encoding: gzip, deflate, br  
Accept: \*/\*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync-collection xmlns="DAV:>
  <sync-token xmlns="DAV:>
    data:,1471360182-7656281
  </sync-token>
  <prop xmlns="DAV:>
    <getetag xmlns="DAV:>"/>
  </prop>
</sync-collection>>
```

**REPORT** /dav/calendars/user/rjbs/my-cool-calendar/ HTTP/1.0

Host: caldav.fastmail.com

Connection: close

Content-Length: 215

Content-Type: text/xml

Depth: 1

Accept-Encoding: gzip, deflate, br

Accept: \*/\*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync-collection xmlns="DAV:">
  <sync-token xmlns="DAV:">
    data:,1471360182-7656281
  </sync-token>
  <prop xmlns="DAV:">
    <getetag xmlns="DAV:"/>
  </prop>
</sync-collection>>
```

**REPORT** /dav/calendars/user/rjbs/my-cool-calendar/ HTTP/1.0

Host: caldav.fastmail.com

Connection: close

Content-Length: 215

Content-Type: text/xml

Depth: 1

Accept-Encoding: gzip, deflate, br

Accept: \*/\*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync-collection xmlns="DAV:">
  <sync-token xmlns="DAV:">
    data:,1471360182-7656281
  </sync-token>
  <prop xmlns="DAV:">
    <getetag xmlns="DAV:"/>
  </prop>
</sync-collection>>
```

You **only** get back what  
changed **since last sync.**

🔥 Inside XML, of course. 🔥



**CardDAV** **davDAV** **CalDAV**



# JMAP

for Calendars



**JMAP**  
for Calendars  
for Contacts



# JMAP

for Calendars  
for Contacts  
for File Storage

request

```
POST /jmap HTTP/1.1
Host: api.fastmail.com
Content-Type: application/json

{
    ...
    "Email/get",
    { "ids": [ "1", "2", "3" ] },
    ...
}
```

request

```
POST /jmap HTTP/1.1
Host: api.fastmail.com
Content-Type: application/json

{
    ...
    "ContactCard/get",
    { "ids": [ "1", "2", "3" ] },
    ...
}
```

Content-Type: application/json

response

```
{ ...
  ...
  "Email/get",
  {
    ...
    "list": [
      {
        ...
        "id": "1",
        "subject": "Hi, Brussels! 🇧🇪❤️",
      }
    ],
  },
  ...
}
```

Content-Type: application/json

response

```
{ ...
  ...
  "ContactCard/get",
  {
    ...
    "list": [
      {
        ...
        "id": "1",
        "name": { "full": "James Ensor" },
        "emails": { ... "jensor@les20.be" ... },
      }
    ],
  },
  ...
}
```

request

```
{ ...
  "Email/changes",
  {
    ...
      "sinceState": "616"
  },
  ...
}
```

response

```
{ ...
  "Email/changes",
  {
    ...
      "oldState": "616",
      "newState": "717",
      "created": [ "a", "b" ],
      "updated": [ "c" ],
      "destroyed": []
  },
  ...
}
```

request

```
{ ...  
  "ContactCard/changes",  
  { ...  
    "sinceState": "616"  
  }, ...  
}
```

response

```
{ ...  
  "ContactCard/changes",  
  { ...  
    "oldState": "616",  
    "newState": "717",  
    "created": [ "a", "b" ],  
    "updated": [ "c" ],  
    "destroyed": []  
  }, ...  
}
```

if you know JMAP for Mail,  
you know the other ones

**it's just new data types**

JMAP for Contacts  
RFC 9610

AddressBook/  
ContactCard/...

JMAP for Calendars  
[draft-ietf-jmap-calendars-26](#)

Calendar/  
CalendarEvent/  
ParticipantIdentity/...

**DAV**

CardDAV collection  
**VCARD**

CalDAV collection  
**VEVENT**  
(unspecified)

**JMAP**

AddressBook/...  
ContactCard/...  
  
Calendar/...  
CalendarEvent/...  
ParticipantIdentity/...

# jMAP for Contacts: ContactCard

# JMAP for Contacts

## a typical ContactCard object

```
{  
    id: "card-123",  
  
    name: { ...name... },  
    addresses: { ...address set... },  
    phones: { ...phone set... },  
    emails: { ...email set... },  
  
    ...  
}
```

# JMAP for Contacts

## a typical ContactCard object's name(s)

```
"name": {  
    "full": "King Ghidorah",  
    "components": [  
        { "kind": "title", "value": "King" },  
        { "kind": "given", "value": "Ghidorah" }  
    ]  
}
```

# JMAP for Contacts

## a typical ContactCard object's name(s)

```
"name": {  
    "full": "King Ghidorah",  
    "components": [  
        { "kind": "title", "value": "King" },  
        { "kind": "given", "value": "Ghidorah" }  
    ]  
}
```



The "kind" values are from Unicode TR35.  
Don't needlessly multiply standards,  
especially ones dealing with human weirdness.

# JMAP for Contacts

## a typical ContactCard object's name(s)

```
"name": {  
    "full": "King Ghidorah",  
    "components": [  
        { "kind": "title", "value": "King" },  
        { "kind": "given", "value": "Ghidorah" }  
    ]  
},  
"nicknames": {  
    "1": { "name": "Monster Zero" }  
}
```

# JMAP for Contacts

## a typical ContactCard object's name(s)

```
"name": {  
    "full": "King Ghidorah",  
    "components": [  
        { "kind": "title", "value": "King" },  
        { "kind": "given", "value": "Ghidorah" }  
    ]  
},  
"nicknames": {  
    "1": { "name": "Monster Zero" }  
},  
"localizations": {  
    "ja": { "name/full": "キングギドラ" }  
}
```

# JMAP for Contacts

a typical ContactCard object's address set

```
{  
  "1": {  
    "contexts": { "work": true },  
    "pref": 1,  
    "components": [  
      { "kind": "name", "value": "4-2-8 Shibakoen"},  
      { "kind": "locality", "value": "Tokyo" }  
    ]  
  }  
}
```

**digression: another way JMAP is great**

```
"addresses": {  
    "1": {  
        "contexts": { "work": true },  
        "pref": 1,  
        "components": [  
            { "kind": "name", "value": "4-2-8 Shibakoen"},  
            { "kind": "locality", "value": "Tokyo" }  
        ]  
    },  
    "xyz": {  
        "contexts": { "private": true },  
        ...  
    }  
}
```

```
"addresses": {  
    "1": {  
        "contexts": { "work": true },  
        "pref": 1,  
        "components": [  
            { "kind": "name", "value": "4-2-8 Shibakoen"},  
            { "kind": "locality", "value": "Tokyo" }  
        ]  
    },  
    "xyz": {  
        "contexts": { "private": true, ... },  
    }  
}
```

**we want to update one, remove one, add one**

# updating a resource in CardDAV

- read the vCard document
- parse it into memory
- update the structure in memory
- turn the whole thing back into vCard
- write it back
- hope you didn't accidentally change anything else while doing so

# updating a resource in JMAP

- decide what property to update
- tell server to apply a patch
- refetch the new state
- know you didn't accidentally change anything else while doing so
- ...and you can use ifInState to avoid weird races

```
"addresses": {  
    "1": {  
        "contexts": { "work": true },  
        "pref": 1,  
        "components": [  
            { "kind": "name", "value": "4-2-8 Shibakoen"},  
            { "kind": "locality", "value": "Tokyo" }  
        ]  
    },  
    "xyz": {  
        "contexts": { "private": true, ... },  
    }  
}
```

```
[  
  "ContactCard/set",  
  {  
    "update": {  
      "card-1": {  
        "addresses/1/pref": 2,  
        "addresses/xyz": null,  
        "addresses/new": { ...whole contact... }  
      }  
    }  
  },  
  ...,  
]
```

**this avoids a lot of pain, trust me**

# JMAP for Calendars: CalendarEvent

# JMAP for Calendars

## a typical CalendarEvent object

```
{  
    "id": "event-123",  
    "title": "Destroy Tokyo",  
    "start": "2027-01-29T03:00:00",  
    "timeZone": "Asia/Tokyo",  
  
    "participants": { ...participant set... },  
    "recurrenceOverrides": { ...override set... },  
    "recurrenceRules": [ ...rrules... ]  
}
```

# JMAP for Calendars

## a typical CalendarEvent object

```
{  
    "id": "event-123",  
    "title": "Destroy Tokyo",  
    "start": "2027-01-29T03:00:00",  
    "timeZone": "Asia/Tokyo",  
  
    "participants": { ...participant set... },  
    "recurrenceOverrides": { ...override set... },  
    "recurrenceRules": [  
        { "frequency": "yearly" }  
    ]  
}
```

# JMAP for Calendars

## a typical CalendarEvent's participants

```
"participants": {  
    "1": {  
        "roles": { "attendee": true },  
        "email": "mayor@tokyo.gov.jp",  
        "participantStatus": "delegated",  
        ...  
    },  
    ...  
}
```

# JMAP for Calendars

## a typical CalendarEvent's overrides

```
"recurrenceOverrides": {  
    "2029-01-29T03:00:00": null,  
    "2038-01-29T03:00:00": {  
        "start": "2038-01-29T08:00:00"  
    }  
}
```

# JMAP for Calendars

## recurrence expansion

**Recurrence rules are a huge pain.**

# JMAP for Calendars

## recurrence expansion

**Recurrence rules are a huge pain.**

Let's say we have an event on the first Saturday of December and the day before it, whatever month that might be.

# JMAP for Calendars

## recurrence expansion

iCalendar

```
FREQ=YEARLY;  
BYYEARDAY=-32,-31,-30,-29,-28,-27,-26,-25;  
BYMONTH=11,12;  
BYDAY=-1FR,1FR,1SA;  
BYSETPOS=1,2
```

# JMAP for Calendars

## recurrence expansion

iCalendar

```
FREQ=YEARLY;  
BYYEARDAY=-32,-31,-30,-29,-28,-27,-26,-25;  
BYMONTH=11,12;  
BYDAY=-1FR,1FR,1SA;  
BYSETPOS=1,2  
  
{  
  "frequency": "yearly",  
  "byYearDay": [ -32, -31, -30, -29, -28, -27, -26, -25 ],  
  "byMonth": [ 11, 12 ],  
  "byDay": [  
    { "day": "fr", "nthOfPeriod": -1 },  
    { "day": "fr", "nthOfPeriod": 1 },  
    { "day": "sa", "nthOfPeriod": 1 }  
  ],  
  "bySetPosition": [ 1, 2 ]  
}
```

JSCalendar

# JMAP for Calendars

## recurrence expansion

```
{ ...
  "Email/query",
  {
    ...
    "filter": [
      {
        ...
        "hasKeyword": "$flagged",
        "from": "rjbs",
      }
    ],
  },
  ...
}
```

# JMAP for Calendars

## recurrence expansion

request

```
{ ...  
  "Email/query",  
  { "filter": { ...filters... } },  
  ...  
}
```

response

```
{ ...  
  "Email/query",  
  { ...  
    "ids": [ "4", "8", "15", "16" ]  
  }, ...  
}
```

# JMAP for Calendars

## recurrence expansion

```
{ ...
  "CalendarEvent/query",
  {
    ...
      "filter": {
        ...
          "after": "2026-01-01T00:00:00",
          "before": "2039-01-01T00:00:00",
          "title": "Destroy"
      }
    ],
  },
}, ...
}
```

# JMAP for Calendars

## recurrence expansion

request

```
...
"CalendarEvent/query", {
  "filter": { ...filters... },
}
```

response

```
...
"CalendarEvent/query", {
  "ids": [ "event-123" ],
},
```

# JMAP for Calendars

## recurrence expansion

request

```
...
"CalendarEvent/query", {
  "filter": { ...filters... },
  "expandRecurrences": true
}
```

response

```
...
"CalendarEvent/query", {
  "ids": [ "event-123-R1", "event-123-R2", ... ],
},
```

JMAP for Contacts  
RFC 9610

AddressBook/...  
ContactCard/...

JMAP for Calendars  
[draft-ietf-jmap-calendars-26](#)

Calendar/...  
CalendarEvent/...  
ParticipantIdentity/...

JMAP for Contacts  
RFC 9610

AddressBook/...  
ContactCard/...

JMAP for Calendars  
[draft-ietf-jmap-calendars-26](#)

Calendar/...  
CalendarEvent/...  
ParticipantIdentity/...

JMAP File Storage  
[draft-ietf-jmap-filenode-00](#)

FileNode/...

**JMAP File Storage is in a very  
early stage of development.**

Fastmail has been using its precursor  
for years, and it's pretty good!

You can follow along with its  
development on the IETF JMAP WG.

# JMAP File Storage

## a typical (boring) FileNode object

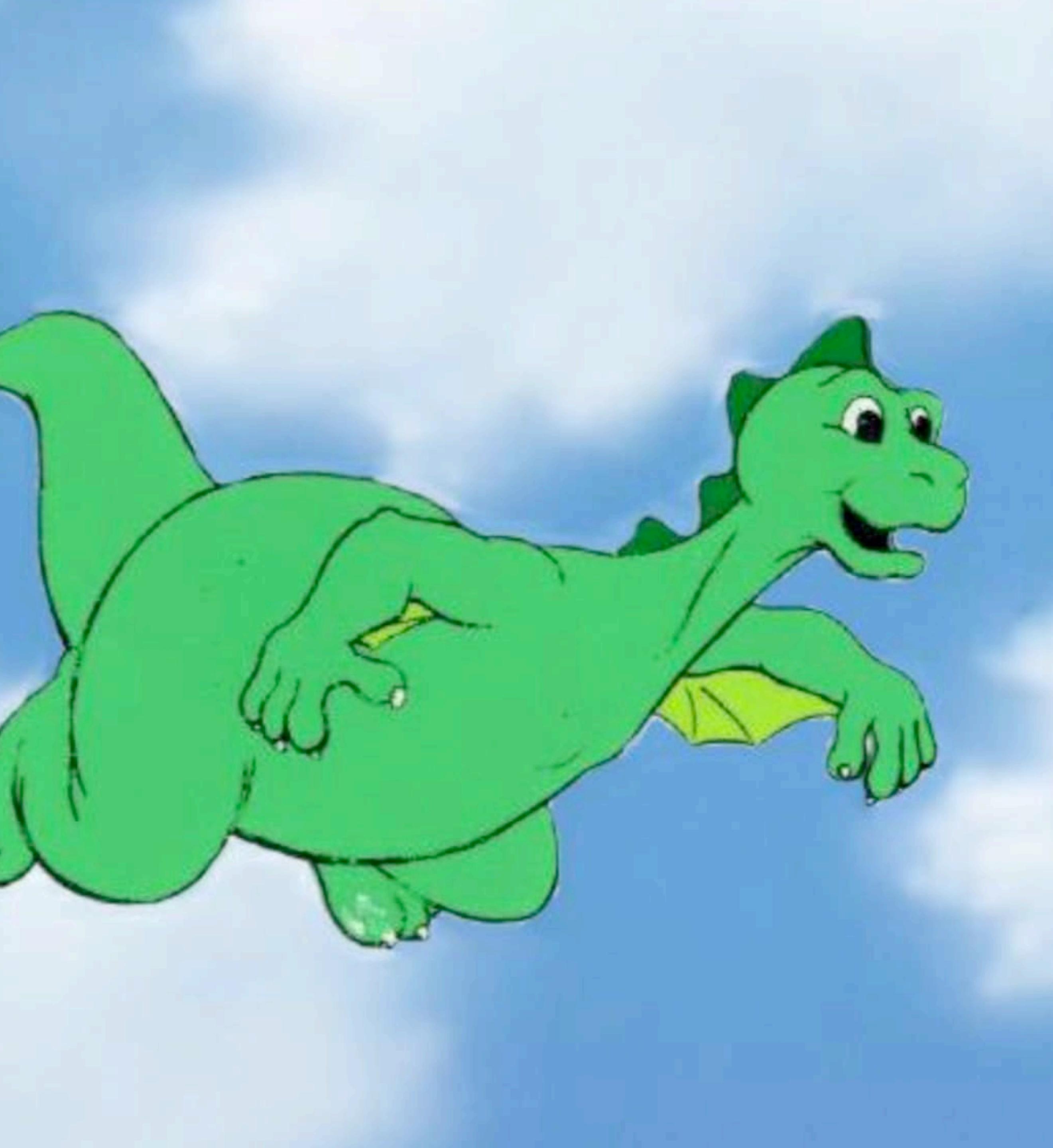
```
{  
    "id": "node-123",  
    "parentId": "node-2",  
    "blobId": "Gxyzzy123",  
    "name": "JMAP the absolute minimum.pdf",  
  
    "executable": false  
}
```

# JMAP File Storage

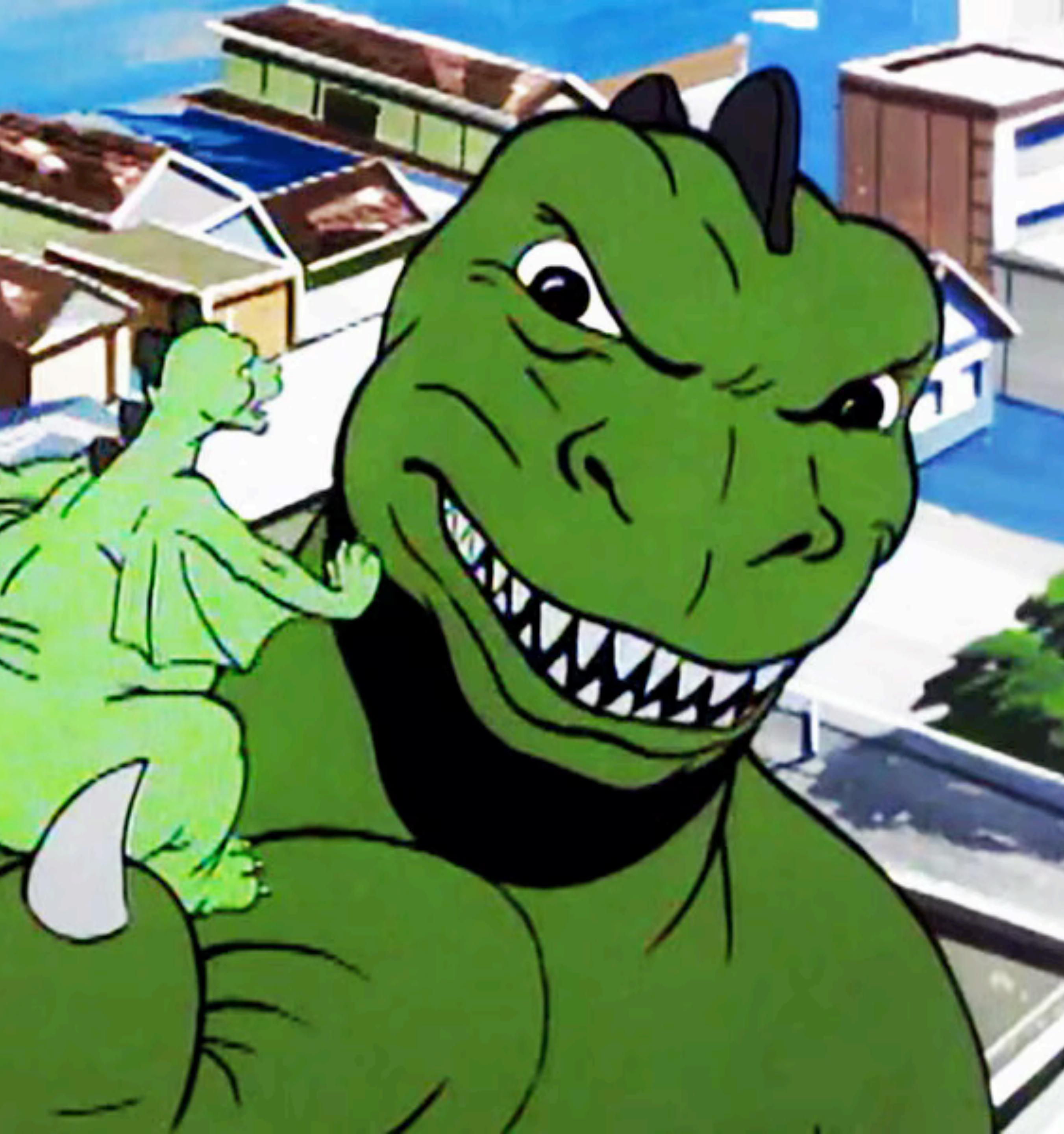
## a typical (boring) FileNode object

```
{  
    "id": "node-123",  
    "parentId": "node-2",  
    "blobId": "Gxyzzy123",  
    "name": "JMAP the absolute minimum.pdf",  
  
    "executable": false  
}
```

JMAP has "store this blob" built in.  
Upload the blob, then make a FileNode that refers to it.

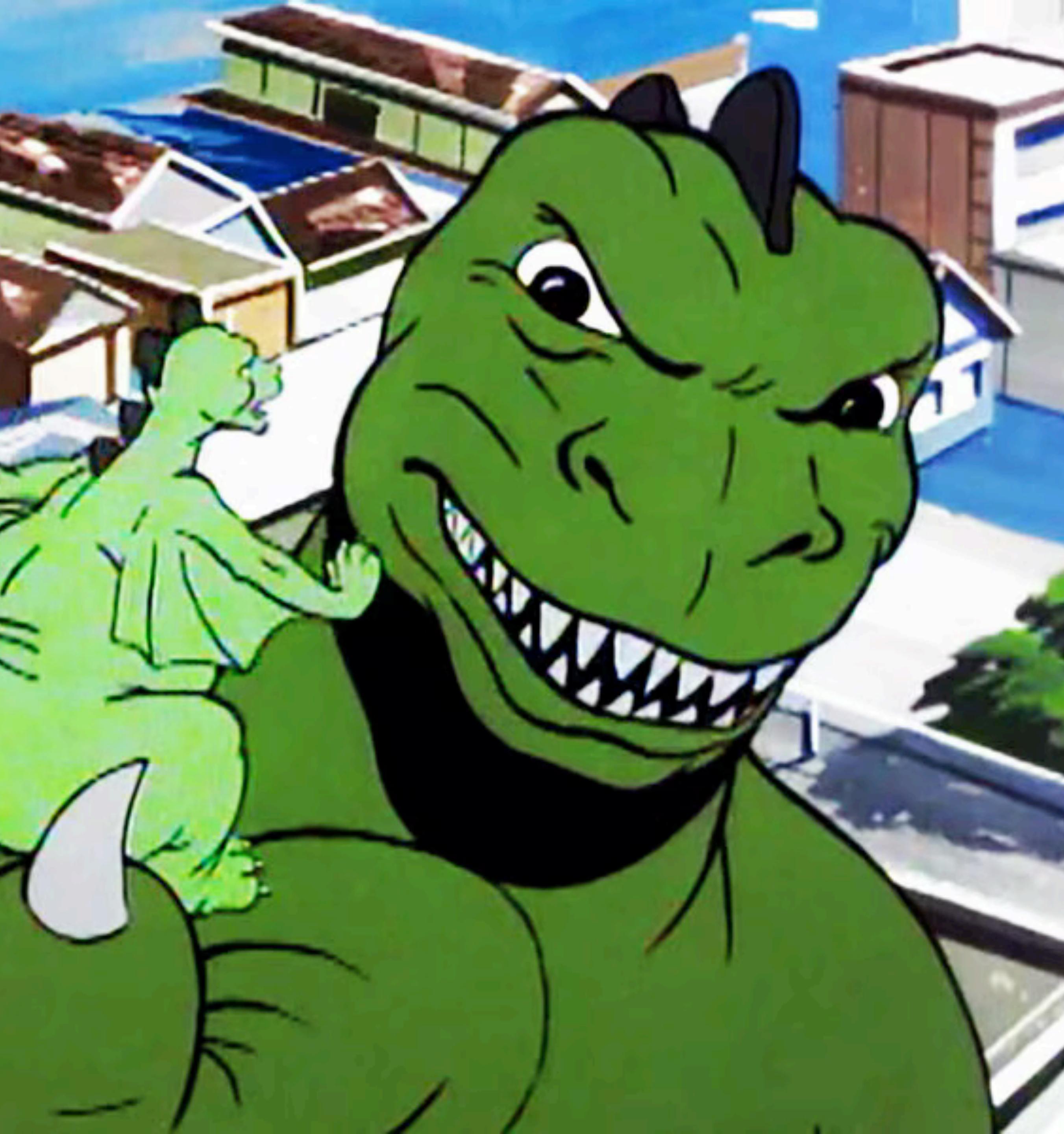


This talk has focused on the basics, so you can get started and play around.



This talk has focused on the basics, so you can get started and play around.

...but JMAP works at scale, and is suitable for Real World use.



This talk has focused on the basics, so you can get started and play around.

...but JMAP works at scale, and is suitable for Real World use.

(It's how Fastmail works!)

# **The Relevant Standards everybody loves reading RFCs**

- RFC 9610: JMAP for Contacts
- RFC 9553: JSContact: A JSON Representation of Contact Data
- RFC 9554: vCard Format Extensions for JSContact
- RFC 9555: JSContact: Converting from and to vCard

# The Relevant Standards everybody loves reading RFCs

- RFC 9610: JMAP for Contacts
- RFC 9553: JSContact: A JSON Representation of Contact Data
- RFC 9554: vCard Format Extensions for JSContact
- RFC 9555: JSContact: Converting from and to vCard

expect updates!

# The Relevant Standards everybody loves reading RFCs

- [draft-ietf-jmap-calendars-26: JMAP for Calendars](#)
- [draft-ietf-calext-jscalendarbis-14: JSCalendar: A JSON Rep'n of Calendar Data](#)
- [draft-ietf-calext-icalendar-jscalendar-extensions-05: iCalendar Extensions](#)
- [draft-ietf-calext-jscalendar-icalendar-22: Converting from and to...](#)

# **The Relevant Standards everybody loves reading RFCs**

- **draft-ietf-jmap-filenode-00: JMAP File Storage**

I know, you're sold!

How can you JMAP?

# Lots of ways!

Ask me more later!



Fastest is to use JMAP on  
your **Fastmail** account\*.

<https://www.fastmail.com/dev/>

\* JMAP for Contacts and Calendars coming real soon.



ANY QUESTIONS?