What's New in UCAN 1.0

Authority Without Boundaries

What's New in UCAN v1.0

Brooklyn Zelenka @expede

What's New in UCAN v1.0

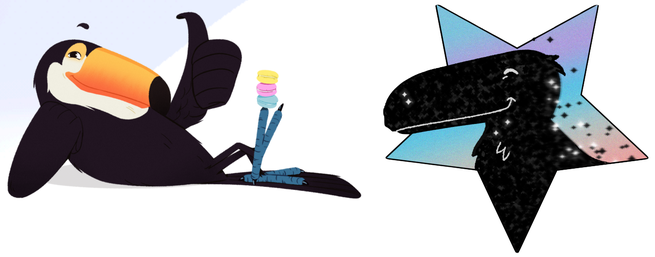
Brooklyn Zelenka @expede

github.com/expede

What's New in UCAN v1.0

Brooklyn Zelenka @expede

- "Open Sourceress" at Witchcraft Software 🧙‍♀️
- Editor of UCAN & IPVM
- UCAN
  - CRDTs, flat file storage, databases, decentralised compute, name systems, email, ...
  - Distributed RPC framework



github.com/expede

What's New in UCAN v1.0

Community

What's New in UCAN v1.0

Community

- 1.0 was a big, community-centred undertaking
- Streamline interfaces & pave the cowpaths!
- Many people in this room contributed in ways big & small 💜
- Takes time & effort to refine & keep as small as possible
  - "Simplicity" is always in service of leverage 🏋️

Crash Course

Crash Course

Everything, Everywhere

Crash Course

Everything, Everywhere

Commons

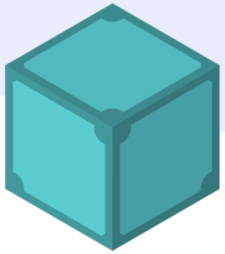
Far Edge

Cloud & Edge

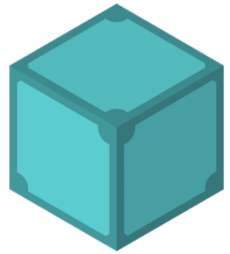
Crash Course

Everything, Everywhere

Commons



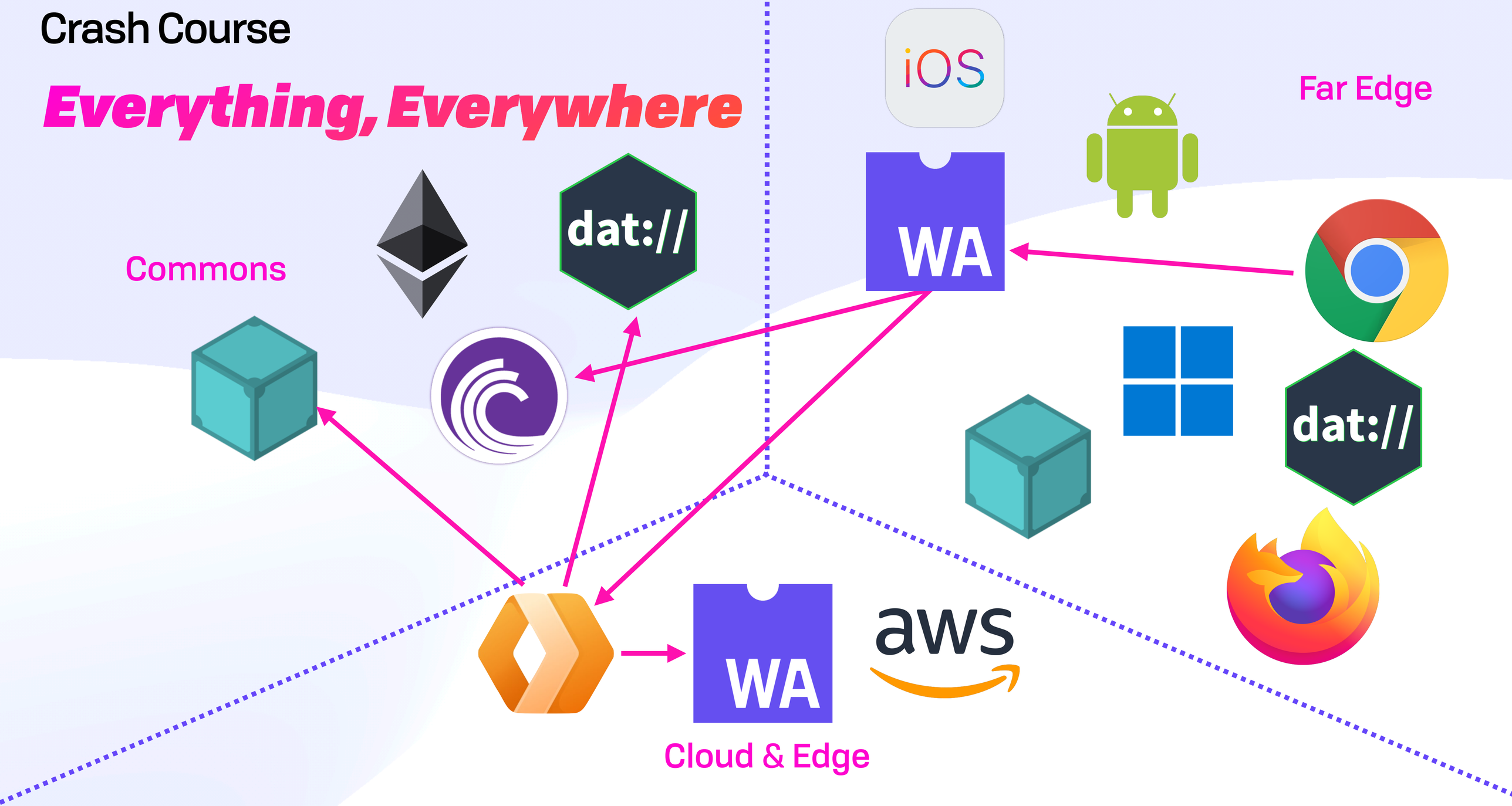
Far Edge



Cloud & Edge

Crash Course

Everything, Everywhere



Crash Course

Certificate Capability Model

Crash Course

Certificate Capability Model

Crash Course

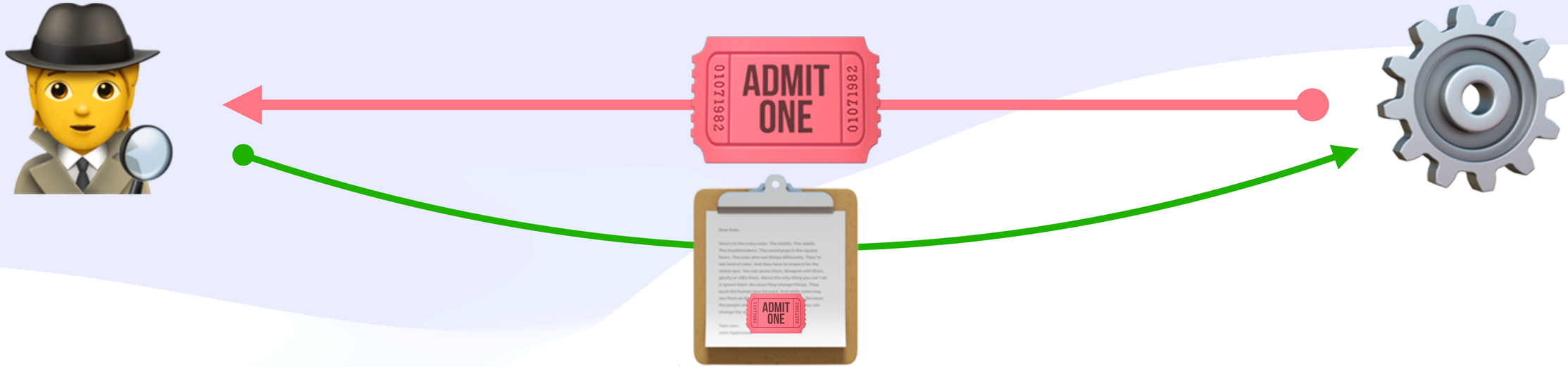
Certificate Capability Model

Crash Course

Certificate Capability Model

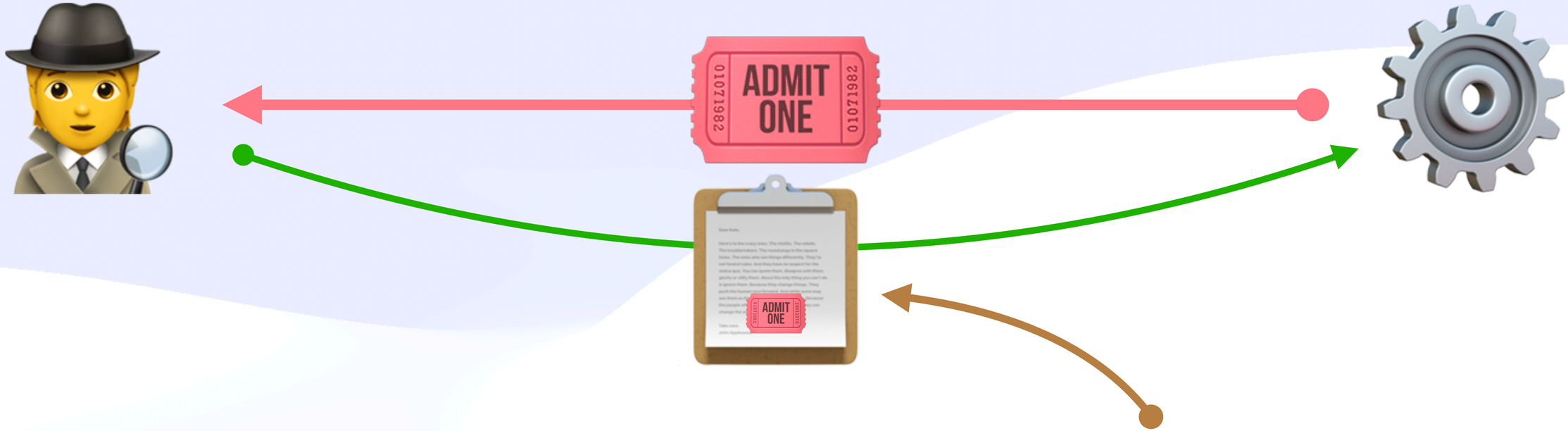
Crash Course

Certificate Capability Model



Crash Course

Certificate Capability Model



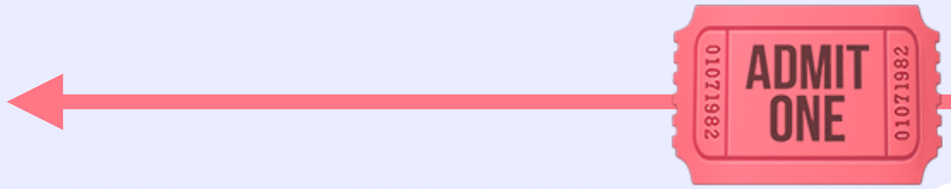
Self-contained:
has all required info

Crash Course

Certificate Capability Model

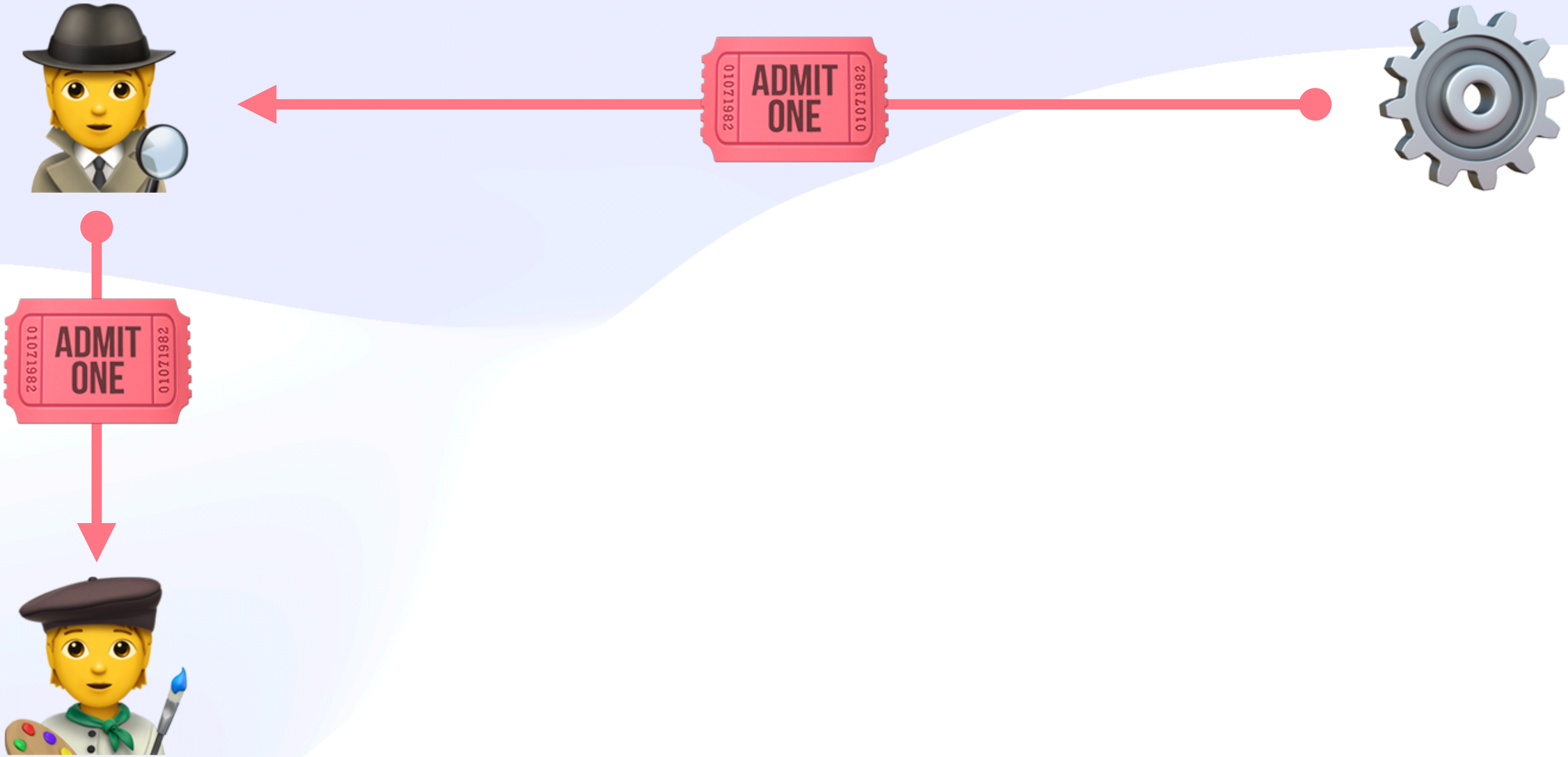
Crash Course

Certificate Capability Model



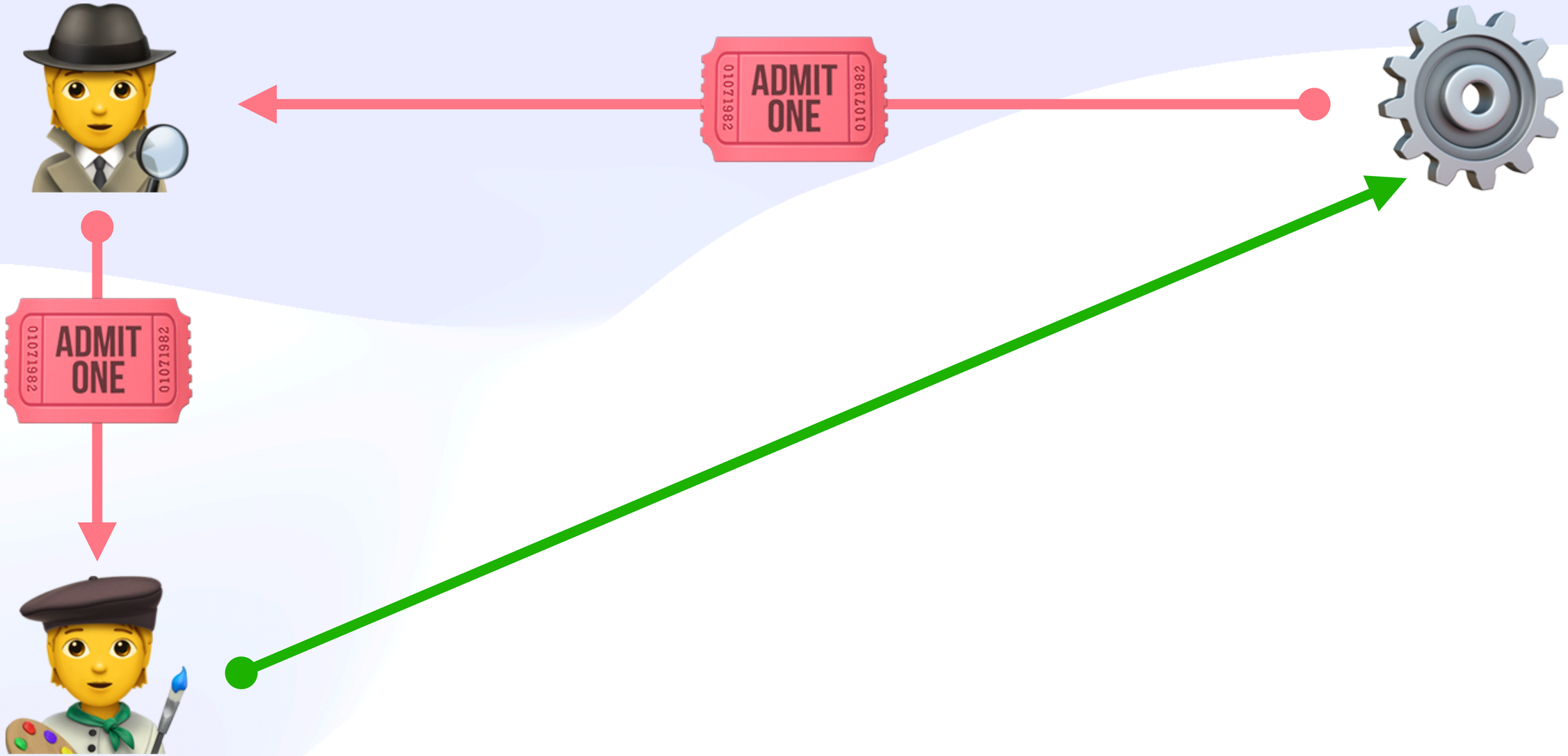
Crash Course

Certificate Capability Model



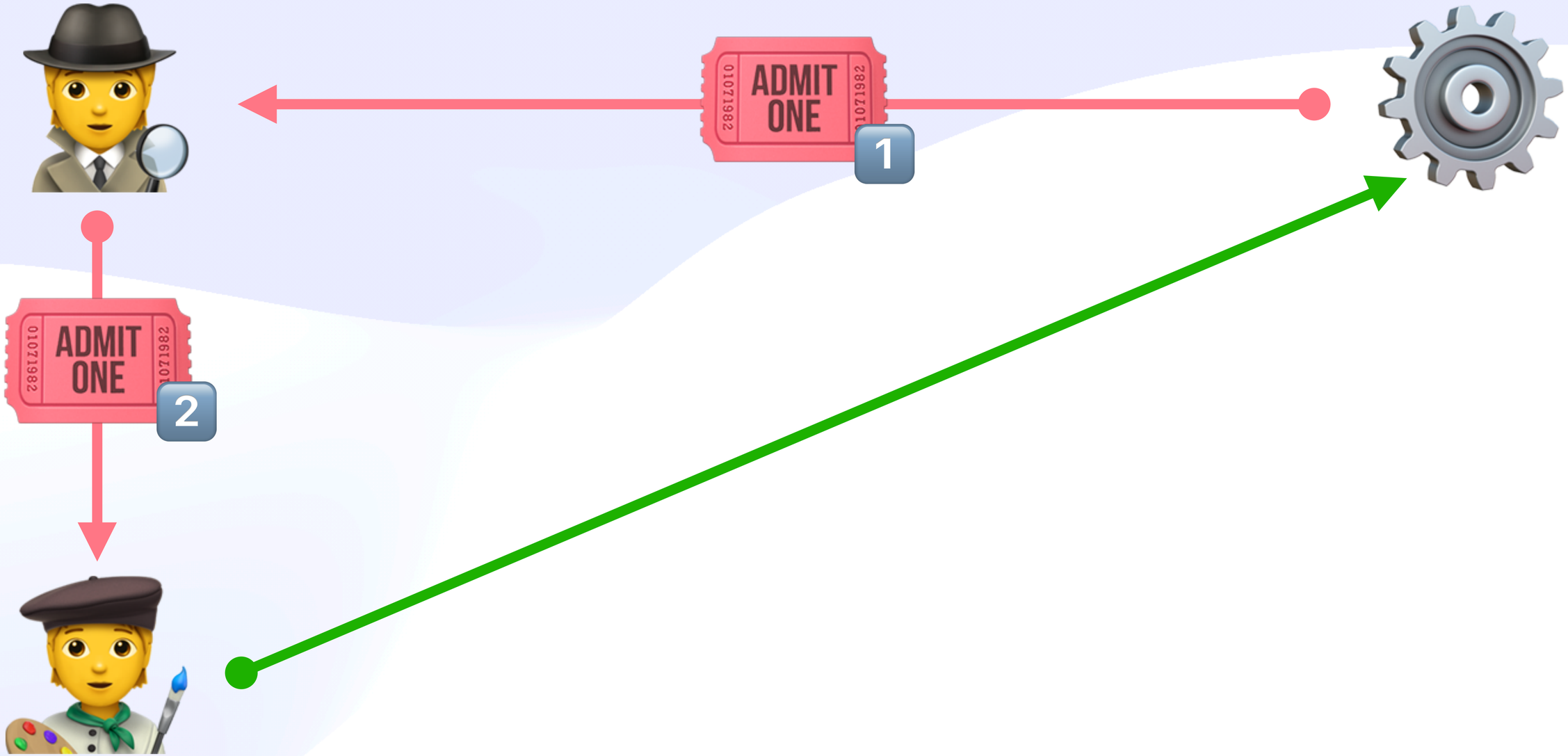
Crash Course

Certificate Capability Model



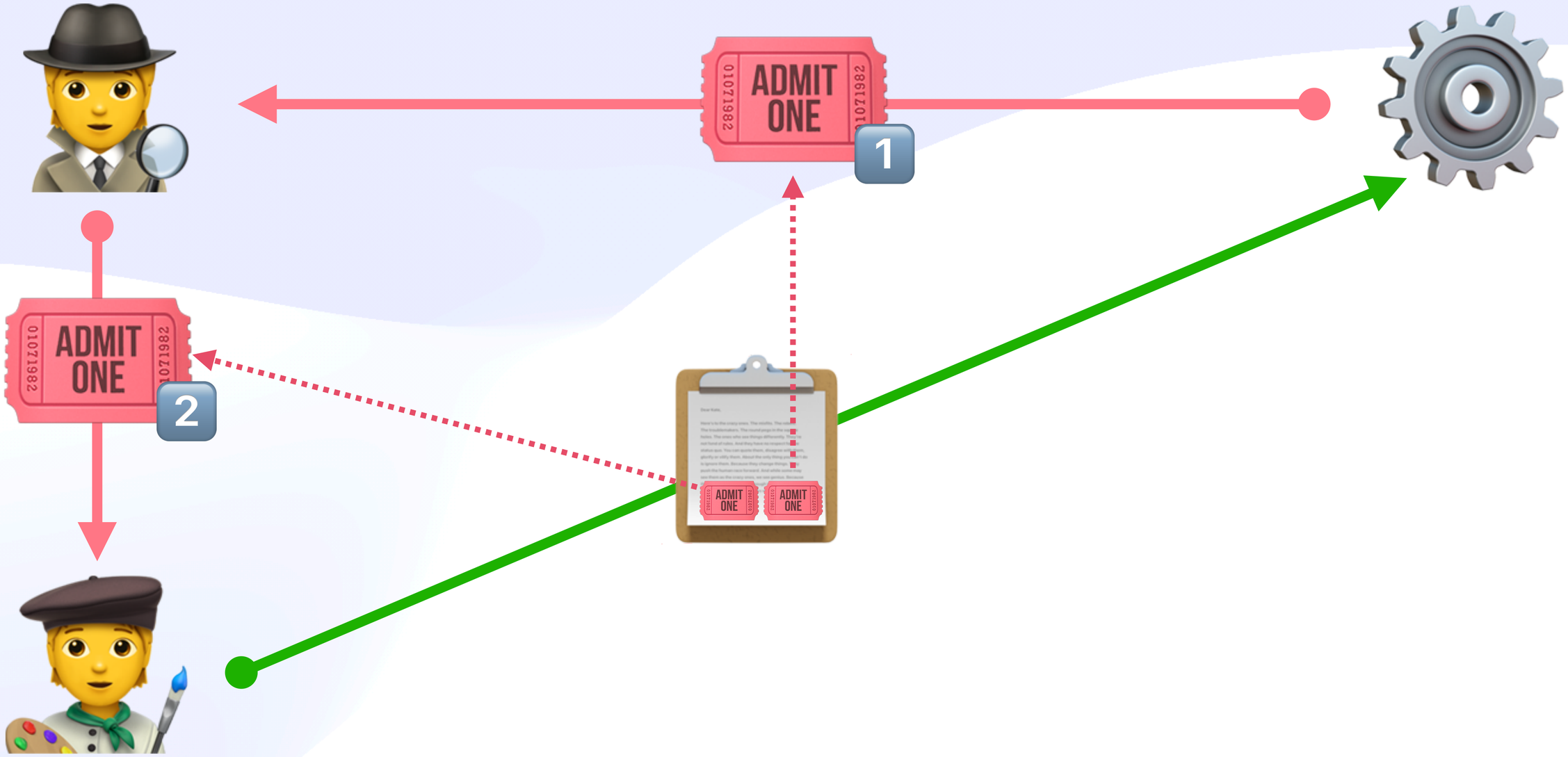
Crash Course

Certificate Capability Model



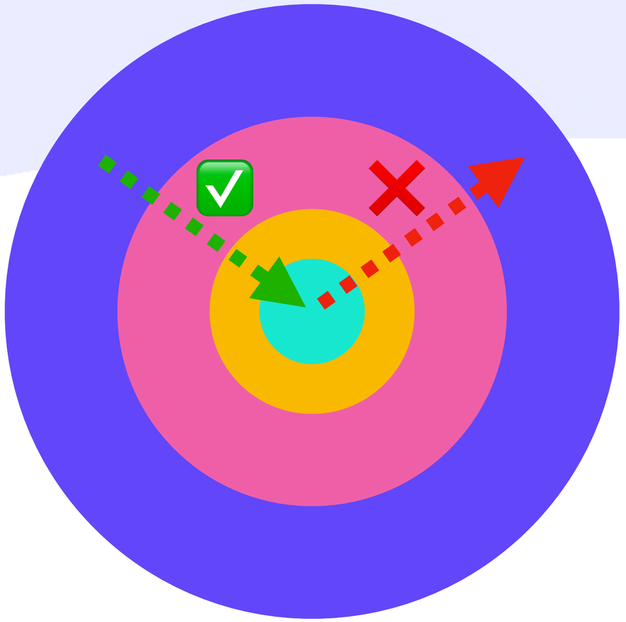
Crash Course

Certificate Capability Model

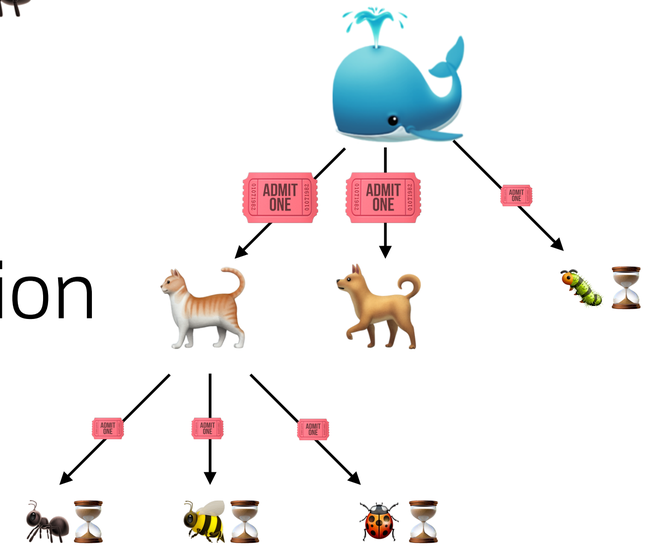


Crash Course

Ad Hoc Caveats

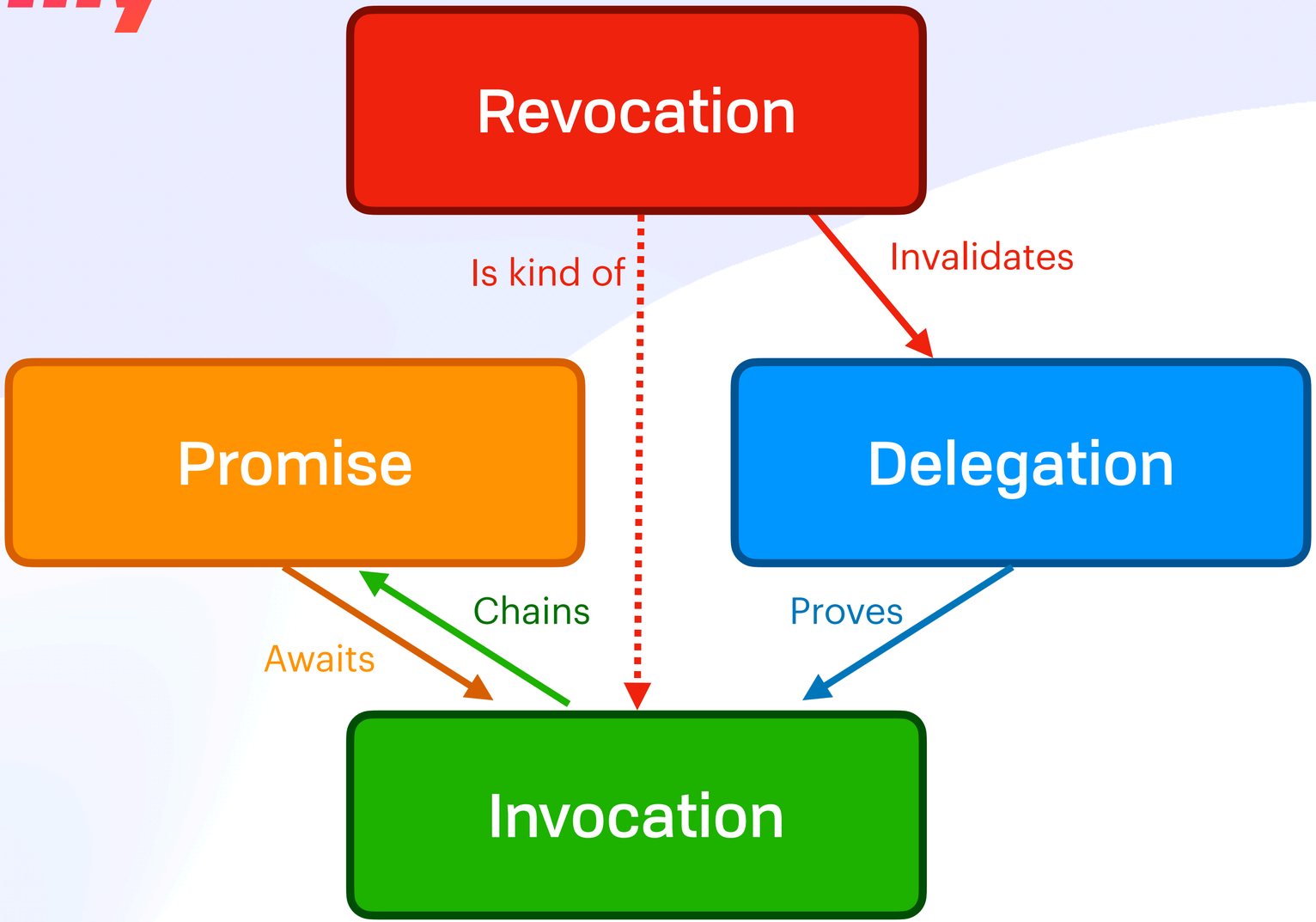


- Agents are **abundant**
  - Few human system administrators 🐋
  - Billions of throwaway single-purpose workers 🐜
- Arbitrarily restrict authority
  - i.e. <u>don't even allow</u> an LLM to hallucinate intention

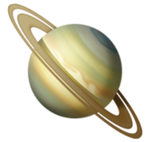


Crash Course

Taxonomy



Interplanification



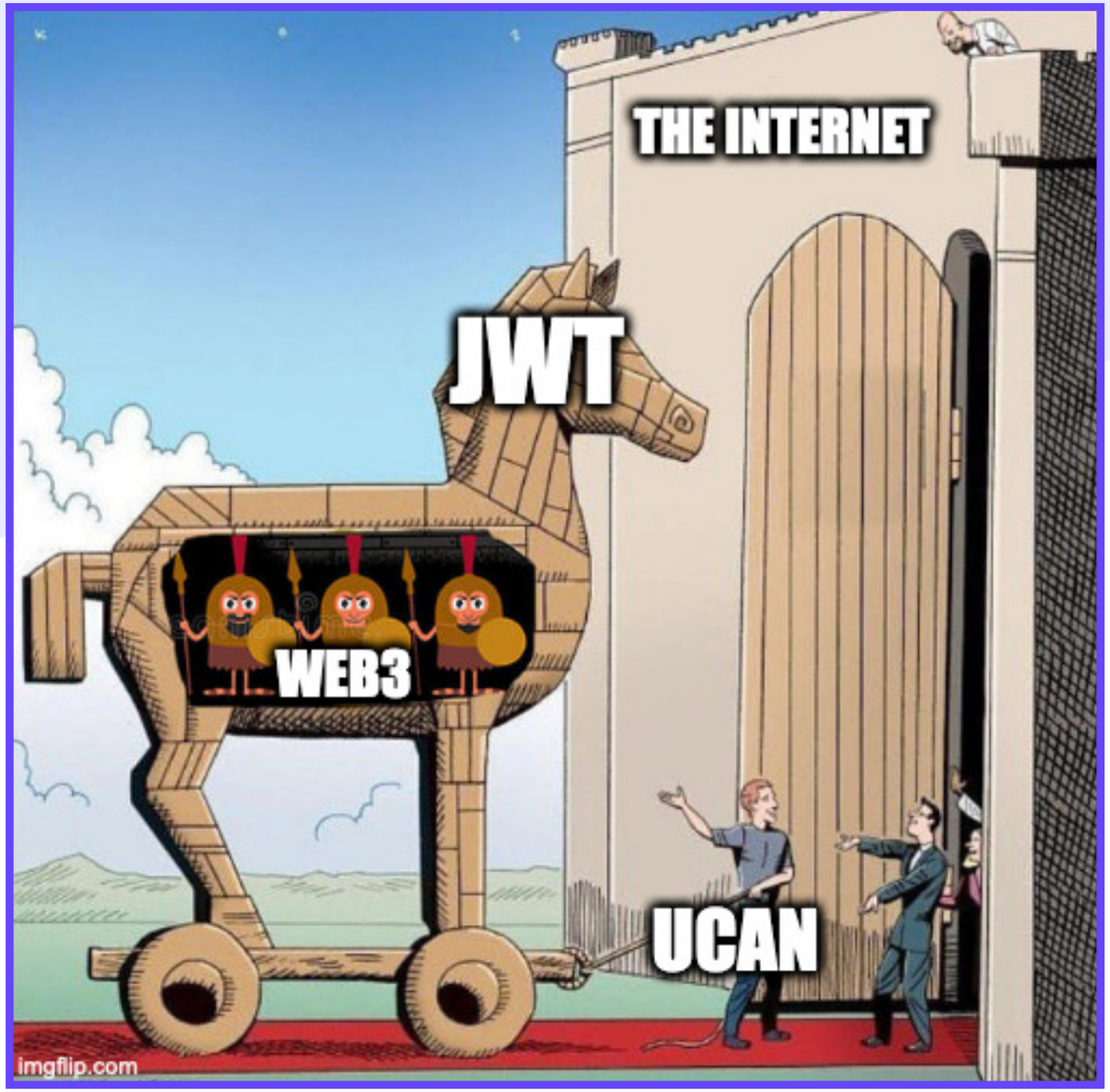
Interplanetification

JWT ➡️ IPLD

- Most UCAN consumers use IPLD
- No more need for `ucan-ipld`!
- JWT "has security issues" if not used "juuuuust so"
- IPLD easier to store, 1st-class support for CIDs, etc

Interplanetification

JWT ➡️ IPLD



- Most UCAN consumers use IPLD
- No more need for `ucan-ipld`!
- JWT "has security issues" if not used "juuuuust so"
- IPLD easier to store, 1st-class support for CIDs, etc

Interplanetification

Canonically DAG-CBOR

- Signature canonicalisation
- Varsig field i.e. the intended way to use self-describing formats

Invocation

Invocation

Concrete (as DAG-JSON)

Invocation

Concrete (as DAG-JSON)

```
{
  "h": {"/": {"bytes": "8i3NboQXu"}},
  "ucan/inv@1.0.0": {
    iss: "did:key:z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLGpbnnEGta2doK",
    aud: "did:key:z6MkiTBz1ymuepAQ4HEHYSF1H8quG5GLVVQR3djdX3mDooWp",

    sub: "did:key:...mailserver",
    cmd: "/email/send",
    args: {
      "to": ["daniel@not.example.com"],
      "cc": ["outreach@example.com", "engineering@example.com"],
      "subject": "Coffee",
      "body": "Still on for coffee in Brussels?"
    },
    nonce: {"/": {"bytes": "y6sBk0_2"}},

    prf: [
      {"/": "bafk...prf1"},
      {"/": "bafk...prf2"}
    ],
    exp: 1234567890
  }
}
```

Invocation

Concrete (as DAG-JSON)

```
{
  "h": {"/": {"bytes": "8i3NboQXu"}},
  "ucan/inv@1.0.0": {
    iss: "did:key:z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLGpbnnEGta2doK",
    aud: "did:key:z6MkiTBz1ymuepAQ4HEHYSF1H8quG5GLVVQR3djdX3mDooWp",

    sub: "did:key:...mailserver",
    cmd: "/email/send",
    args: {
      "to": ["daniel@not.example.com"],
      "cc": ["outreach@example.com", "engineering@example.com"],
      "subject": "Coffee",
      "body": "Still on for coffee in Brussels?"
    },
    nonce: {"/": {"bytes": "y6sBk0_2"}},

    prf: [
      {"/": "bafk...prf1"},
      {"/": "bafk...prf2"}
    ],
    exp: 1234567890
  }
}
```

Invocation

Command Standard Library

Invocation

Command Standard Library

```
/crud/create
/crud/read
/crud/update
/crud/destroy

/msg/send
/msg/receive

/group/ban
/group/join
```

Invocation

Command Standard Library

```
/crud/create
/crud/read
/crud/update
/crud/destroy

/msg/send
/msg/receive

/group/ban
/group/join
```

Determines shape

```
{
  to: ["mailto:alice@example.com"],
  subject: null,
  body: "hello world"
}
```

Invocation

Command Standard Library

```
/crud/create
/crud/read
/crud/update
/crud/destroy

/msg/send
/msg/receive

/group/ban
/group/join
```

```
/ucan/revoke
```

Determines shape

```
{
  to: ["mailto:alice@example.com"],
  subject: null,
  body: "hello world"
}
```

Invocation: Revocation

Accidents Happen 🤷‍♀️

Invocation: Revocation

Accidents Happen 🤷‍♀️

Byzantine Agent

Invocation: Revocation

Accidents Happen 🤷‍♀️

Byzantine Agent

Revocation

Invocation: Revocation

Eventually Consistent Revocation

```
{
  iss: "did:key:...bob...",
  aud: "did:key:...alice...",
  sub: "did:key:...alice..."
  cmd: "/ucan/revoke",
  args: {
    chain: [
      {"/": "bafy_bob_to_carol"},
      {"/": "bafy_carol_to_mallory"}
    ]
  },
  prf: []
  // ...
}
```

Invocation: Revocation

Eventually Consistent Revocation

```
{
  iss: "did:key:...bob...",
  aud: "did:key:...alice...",
  sub: "did:key:...alice..."
  cmd: "/ucan/revoke",
  args: {
    chain: [
      {"/": "bafy_bob_to_carol"},
      {"/": "bafy_carol_to_mallory"}
    ]
  },
  prf: []
  // ...
}
```

Invocation: Revocation

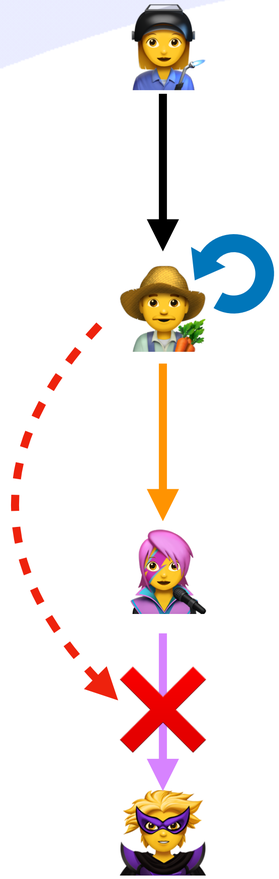
Eventually Consistent Revocation

```
{
  iss: "did:key:...bob...",
  aud: "did:key:...alice...",
  sub: "did:key:...alice..."
  cmd: "/ucan/revoke",
  args: {
    chain: [
      {"/": "bafy_bob_to_carol"},
      {"/": "bafy_carol_to_mallory"}
    ]
  },
  prf: []
  // ...
}
```

Invocation: Revocation

Eventually Consistent Revocation

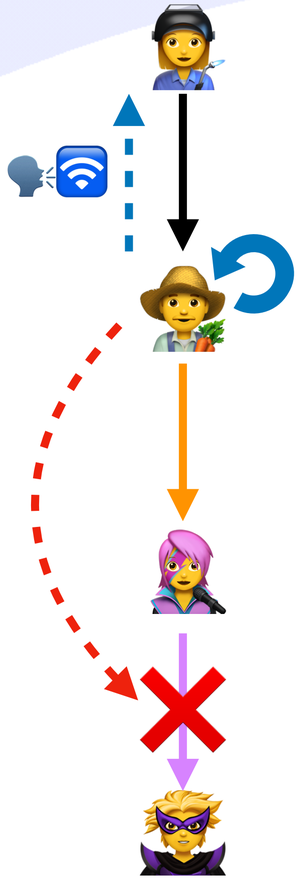
```
{
  iss: "did:key:...bob...",
  aud: "did:key:...alice...",
  sub: "did:key:...alice..."
  cmd: "/ucan/revoke",
  args: {
    chain: [
      {"/": "bafy_bob_to_carol"},
      {"/": "bafy_carol_to_mallory"}
    ]
  },
  prf: []
  // ...
}
```



Invocation: Revocation

Eventually Consistent Revocation

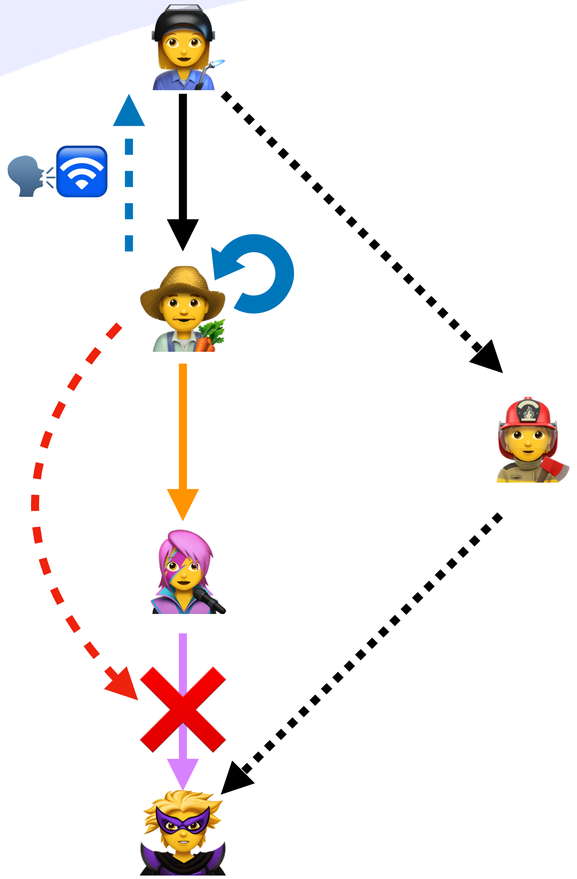
```
{
  iss: "did:key:...bob...",
  aud: "did:key:...alice...",
  sub: "did:key:...alice..."
  cmd: "/ucan/revoke",
  args: {
    chain: [
      {"/": "bafy_bob_to_carol"},
      {"/": "bafy_carol_to_mallory"}
    ]
  },
  prf: []
  // ...
}
```



Invocation: Revocation

Eventually Consistent Revocation

```
{
  iss: "did:key:...bob...",
  aud: "did:key:...alice...",
  sub: "did:key:...alice..."
  cmd: "/ucan/revoke",
  args: {
    chain: [
      {"/": "bafy_bob_to_carol"},
      {"/": "bafy_carol_to_mallory"}
    ]
  },
  prf: []
  // ...
}
```



Invocation: Promises

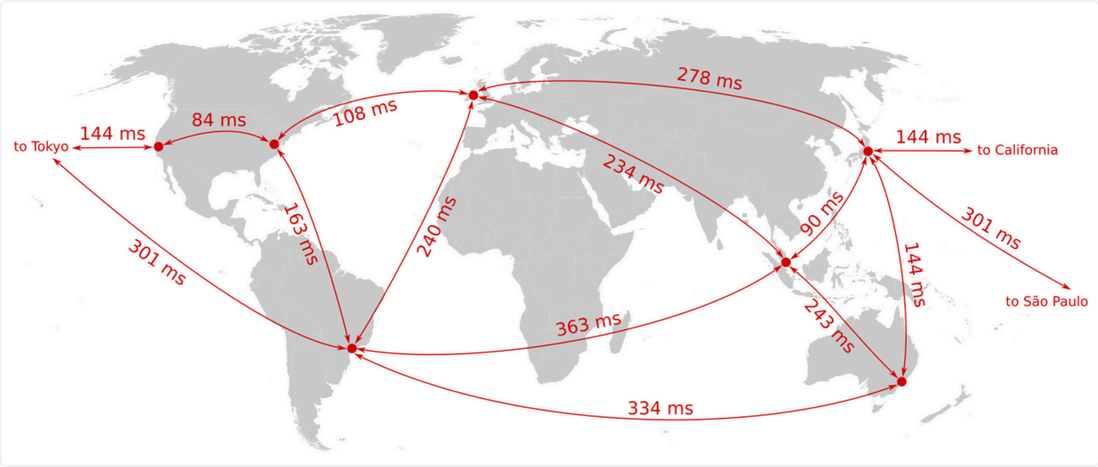
Promises, Promises 😵

Invocation: Promises

Promises, Promises 😲

1. No spinners: your work at your fingertips

Much of today's software feels slower than previous generations of software. Even though CPUs have become ever faster, there is often a perceptible delay between some user input (e.g. clicking a button, or hitting a key) and the corresponding result appearing on the display. In previous work we measured the performance of modern software and analyzed why these delays occur.

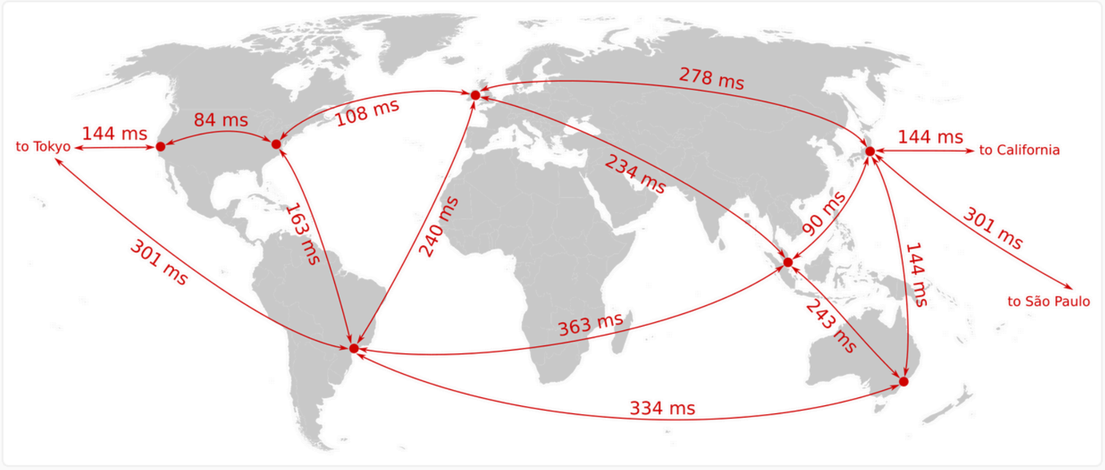


Invocation: Promises

Promises, Promises 😲

1. No spinners: your work at your fingertips

Much of today's software feels slower than previous generations of software. Even though CPUs have become ever faster, there is often a perceptible delay between some user input (e.g. clicking a button, or hitting a key) and the corresponding result appearing on the display. In previous work we measured the performance of modern software and analyzed why these delays occur.



New York is not getting any closer to Tokyo […]
The latency barrier […] will increasingly dominate

— Mark Miller, Robust Composition

Invocation: Promises

Input Addressing — "TaskID"

Invocation: Promises

Input Addressing — "TaskID"

```
{
  "h": {"/": {"bytes": "8i3NboQXu"}},
  "ucan/inv@1.0.0": {
    iss: "did:key:z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLGpbnnEGta2doK",
    aud: "did:key:z6MkiTBz1ymuepAQ4HEHYSF1H8quG5GLVVQR3djdX3mDooWp",

    sub: "did:key:...mailserver",
    cmd: "/email/send",
    args: {
      "to": ["daniel@not.example.com"],
      "cc": ["outreach@example.com", "engineering@example.com"],
      "subject": "Coffee",
      "body": "Still on for coffee in Brussels?"
    },
    nonce: {"/": {"bytes": "y6sBk0_2"}},

    prf: [
      {"/": "bafk...prf1"},
      {"/": "bafk...prf2"}
    ],
    exp: 1234567890
  }
}
```

Invocation: Promises

Input Addressing — "TaskID"

```
{
  "h": {"/": {"bytes": "8i3NboQXu"}},
  "ucan/inv@1.0.0": {
    iss: "did:key:z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLGpbnnEGta2doK",
    aud: "did:key:z6MkiTBz1ymuepAQ4HEHYSF1H8quG5GLVVQR3djdX3mDooWp",
    sub: "did:key:...mailserver",
    cmd: "/email/send",
    args: {
      "to": ["daniel@not.example.com"],
      "cc": ["outreach@example.com", "engineering@example.com"],
      "subject": "Coffee",
      "body": "Still on for coffee in Brussels?"
    },
    nonce: {"/": {"bytes": "y6sBk0_2"}},
    prf: [
      {"/": "bafk...prf1"},
      {"/": "bafk...prf2"}
    ],
    exp: 1234567890
  }
}
```

Input uniquely identifies a (semantic) "task"

Invocation: Promises

Input Addressing — "TaskID"

Invocation: Promises

Input Addressing — "TaskID"

```
{
  "h": {"/": {"bytes": "8i3NboQXu"}},
  "ucan/inv@1.0.0": {
    iss: "did:key:z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLGpbnnEGta2doK",
    aud: "did:key:z6MkiTBz1ymuepAQ4HEHYSF1H8quG5GLVVQR3djdX3mDooWp",

    sub: "did:key:...mailserver",
    cmd: "/email/send",
    args: {
      "to": ["daniel@not.example.com"],
      "cc": ["outreach@example.com", "engineering@example.com"],
      "subject": {"ucan/await": [".ok", {"/": "bafk...other"}]},
      "body": "Still on for coffee in Brussels?"
    },
    nonce: {"/": {"bytes": "y6sBk0_2"}},

    prf: [
      {"/": "bafk...prf1"},
      {"/": "bafk...prf2"}
    ],
    exp: 1234567890
  }
}
```

Invocation: Promises

Input Addressing — "TaskID"

```
{
  "h": {"/": {"bytes": "8i3NboQXu"}},
  "ucan/inv@1.0.0": {
    iss: "did:key:z6MkhaXgBZDvotDkL5257faiztiGiC2QtKLGpbnnEGta2doK",
    aud: "did:key:z6MkiTBz1ymuepAQ4HEHYSF1H8quG5GLVVQR3djdX3mDooWp",

    sub: "did:key:...mailserver",
    cmd: "/email/send",
    args: {
      "to": ["daniel@not.example.com"],
      "cc": ["outreach@example.com", "engineering@example.com"],
      "subject": {"ucan/await": [".ok", {"/": "bafk...other"}]},
      "body": "Still on for coffee in Brussels?"
    },
    nonce: {"/": {"bytes": "y6sBk0_2"}},

    prf: [
      {"/": "bafk...prf1"},
      {"/": "bafk...prf2"}
    ],
    exp: 1234567890
  }
}
```

Invocation: Promises

Abstract RPC = Protocol Language

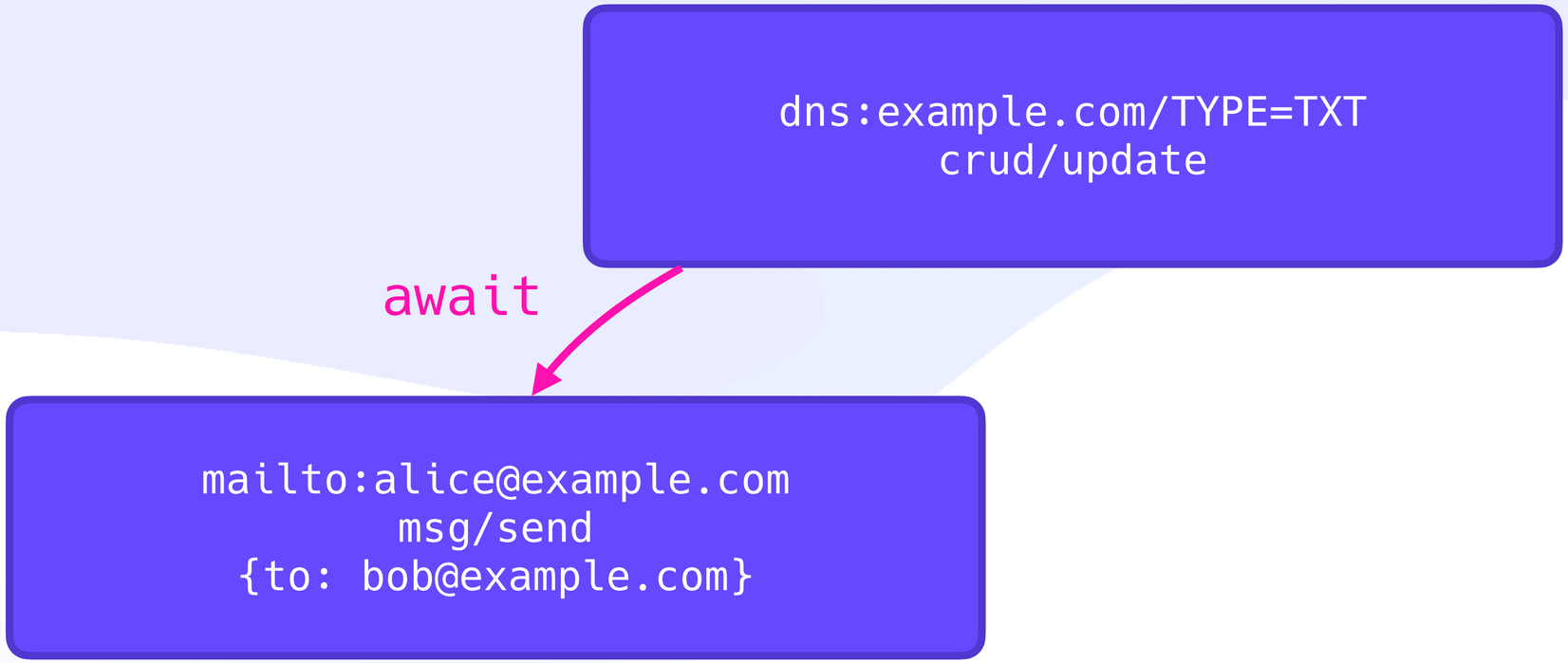
Invocation: Promises

Abstract RPC = Protocol Language

```
dns:example.com/TYPE=TXT
crud/update
```

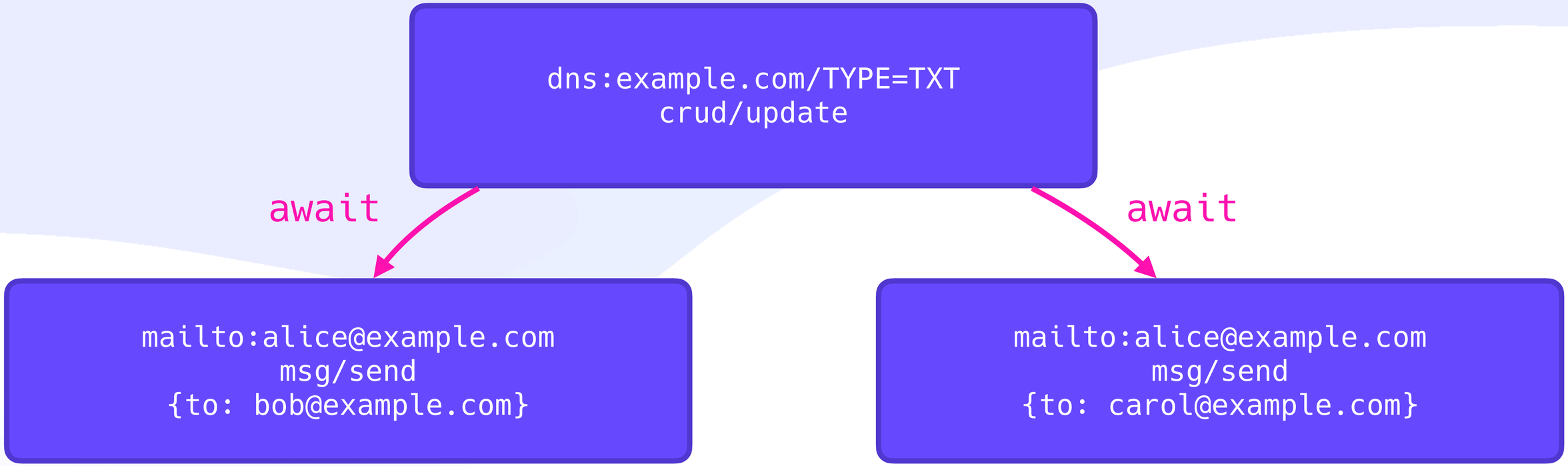
Invocation: Promises

Abstract RPC = Protocol Language



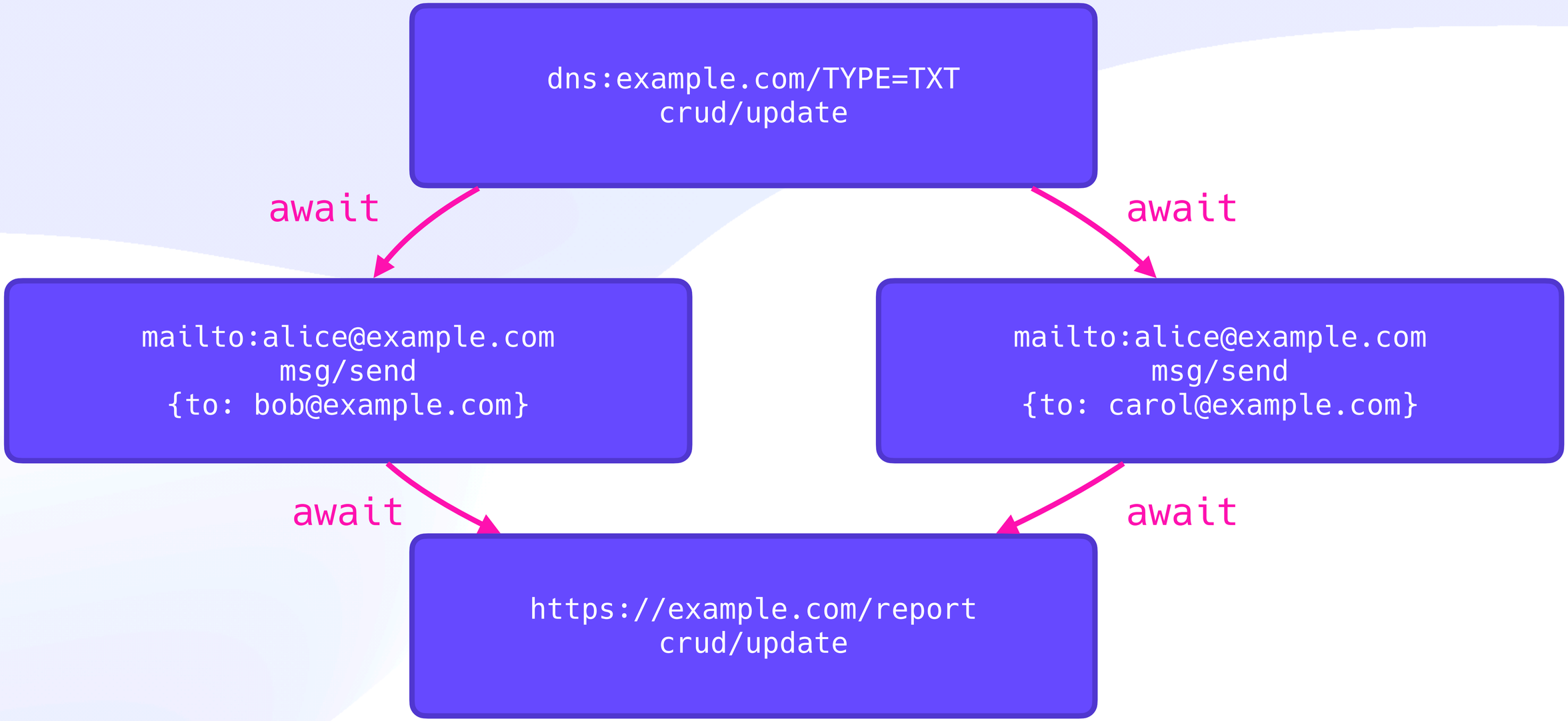
Invocation: Promises

Abstract RPC = Protocol Language



Invocation: Promises

Abstract RPC = Protocol Language



Delegation

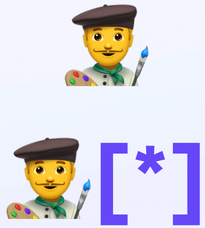
Raw Mechanics

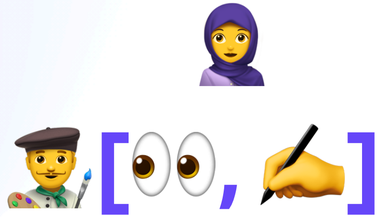
Checking Chains

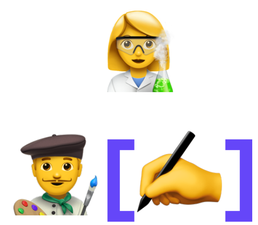
About: 👨‍🎨
From: 👨‍🎨
To: 🧕
Can: [👀, ✍️]

About: 👨‍🎨
From: 🧕
To: 👩‍🎤
Can: [👀,✍️]

About: 👨‍🎨
From: 👩‍🎤
To: 👰
Can: [✍️]

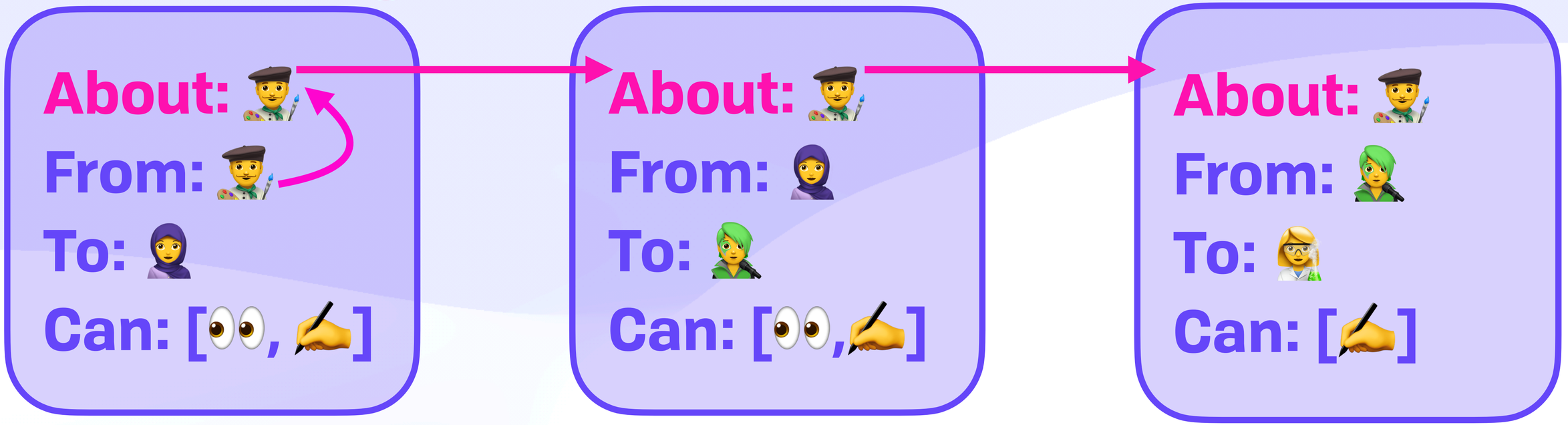






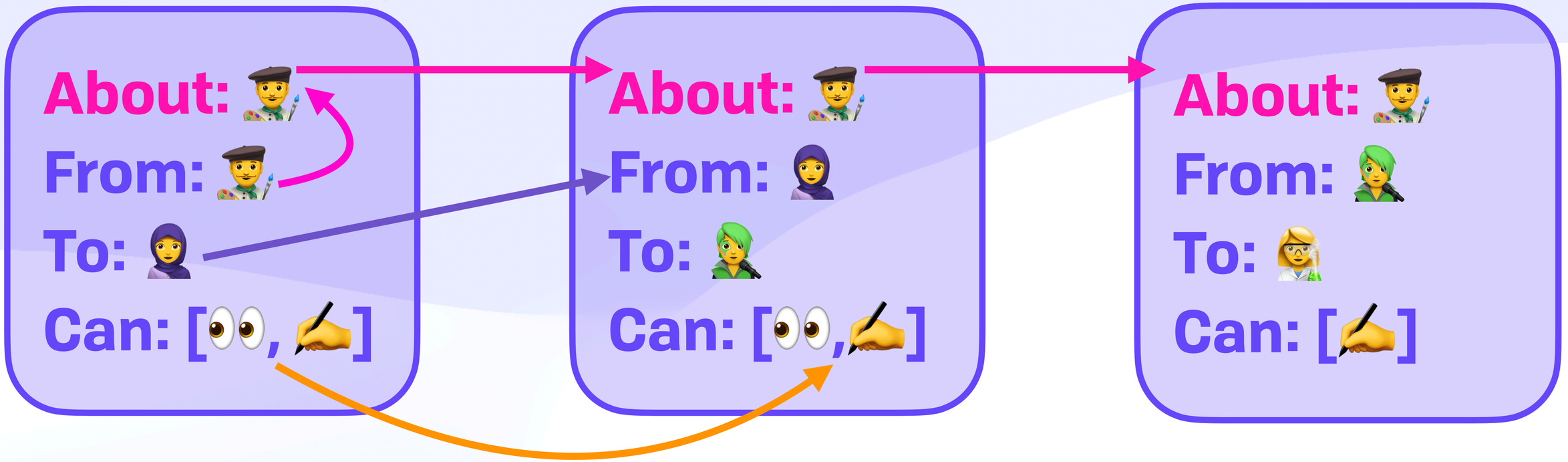
Raw Mechanics

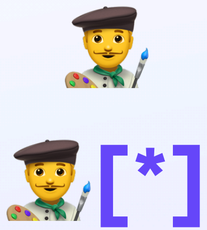
Checking Chains

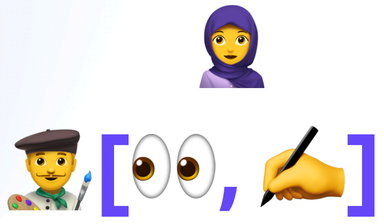


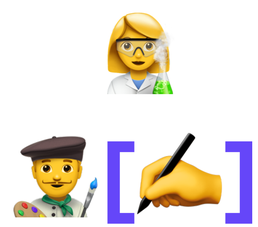
Raw Mechanics

Checking Chains



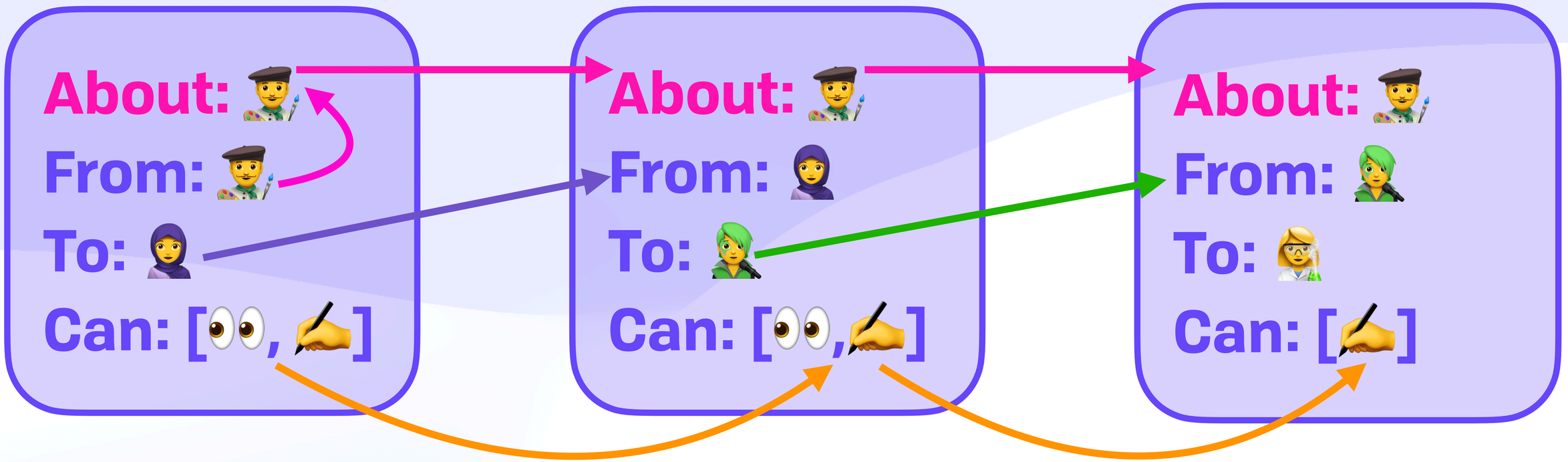


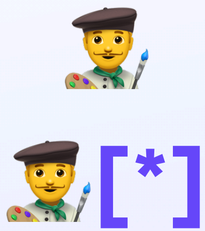


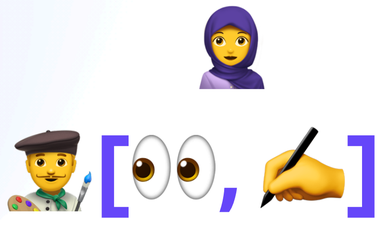


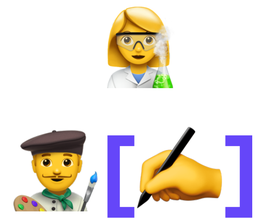
Raw Mechanics

Checking Chains









Invocation: Promises

Policy Language

```
["==", ".planet.name", "Saturn"]
```

```
[
  "and", [
    [">=", ".team[3]?.size", 4],
    ["==", ".ceo.first_name", "Juan"]
  ]
]
```

```
[
  "every", ".recipient", [
    "or", [
      ["match", ".email", "*@example.com"],
      ["==",    ".email", "fraud@not.example.com"]
    ]
  ]
]
```

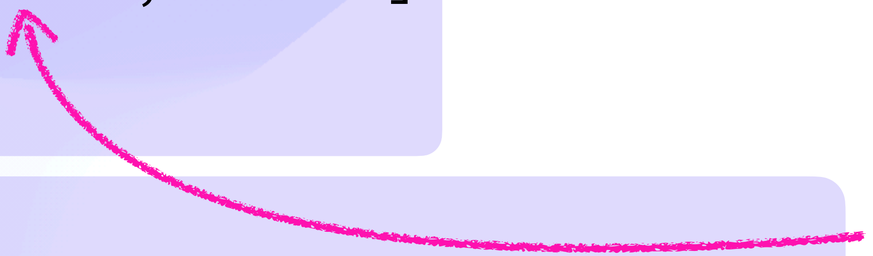
Invocation: Promises

Policy Language

```
["==", ".planet.name", "Saturn"]
```

```
[
  "and", [
    [">=", ".team[3]?.size", 4],
    ["==", ".ceo.first_name", "Juan"]
  ]
]
```

```
[
  "every", ".recipient", [
    "or", [
      ["match", ".email", "*@example.com"],
      ["==",    ".email", "fraud@not.example.com"]
    ]
  ]
]
```



jq-style selectors

Invocation: Promises

Policy Language

```
["==", ".planet.name", "Saturn"]
```

```
[
  "and", [
    [">=", ".team[3]?.size", 4],
    ["==", ".ceo.first_name", "Juan"]
  ]
]
```

```
[
  "every", ".recipient", [
    "or", [
      ["match", ".email", "*@example.com"],
      ["==",    ".email", "fraud@not.example.com"]
    ]
  ]
]
```

```
==, match   not   every
>, ⩾       and    some
<, ⩽        or
```

jq-style selectors

Raw Mechanics

Syntactically Driven

```
{
  iss: "did:key:alice",
  aud: "did:key:bob",
  sub: "did:key:doc"
  cmd: "/automerge/update",
  pol: [
    ["gt", ".foo[1].bar", "42"],
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]
  ]
}
```

```
{
  iss: "did:key:bob",
  aud: "did:key:doc",
  sub: "did:key:doc"
  prf: [hash(delegation), hash(more_chain)]
  cmd: "/automerge/update",
  args: {
    foo: [
      {bar: 100}, // > 42
      {quux: "unconstrained"}
    ],
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4
  }
}
```

Raw Mechanics

Syntactically Driven

```
{
  iss: "did:key:alice",
  aud: "did:key:bob",
  sub: "did:key:doc"
  cmd: "/automerge/update",
  pol: [
    ["gt", ".foo[1].bar", "42"],
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]
  ]
}
```

```
{
  iss: "did:key:bob",
  aud: "did:key:doc",
  sub: "did:key:doc"
  prf: [hash(delegation), hash(more_chain)]
  cmd: "/automerge/update",
  args: {
    foo: [
      {bar: 100}, // > 42
      {quux: "unconstrained"}
    ],
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4
  }
}
```

Raw Mechanics

Syntactically Driven

```
{
  iss: "did:key:alice",
  aud: "did:key:bob",
  sub: "did:key:doc"
  cmd: "/automerge/update",
  pol: [
    ["gt", ".foo[1].bar", "42"],
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]
  ]
}
```

```
{
  iss: "did:key:bob",
  aud: "did:key:doc",
  sub: "did:key:doc"
  prf: [hash(delegation), hash(more_chain)]
  cmd: "/automerge/update",
  args: {
    foo: [
      {bar: 100}, // > 42
      {quux: "unconstrained"}
    ],
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4
  }
}
```

Raw Mechanics

Syntactically Driven

```
{
  iss: "did:key:alice",
  aud: "did:key:bob",
  sub: "did:key:doc"
  cmd: "/automerge/update",
  pol: [
    ["gt", ".foo[1].bar", "42"],
    ["eq", ".prev", 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4]
  ]
}
```

```
{
  iss: "did:key:bob",
  aud: "did:key:doc",
  sub: "did:key:doc"
  prf: [hash(delegation), hash(more_chain)]
  cmd: "/automerge/update",
  args: {
    foo: [
      {bar: 100}, // > 42
      {quux: "unconstrained"}
    ],
    prev: 0x98ea6e4f216f2fb4b69fff9b3a44842c38686ca685f3f55dc48c5d3fb1107be4
  }
}
```

Yes, UCAN!

Wrap Up

Wrap Up

v1.0 Implementations

- rs-ucan v0.5
  - `https://github.com/expede/rs-ucan`
  - Native Rust works today, WebAssembly "very soon"
- ucanto (JavaScript UCAN-based RPC framework)
  - `https://github.com/web3-storage/ucanto`
  - Storacha updating ucanto to latest spec ❤️‍🔥

Wrap Up

Join Us!

github.com/ucan-wg

lu.ma/wecan

github.com/ucan-wg

lu.ma/wecan

🎉 Thank You, IPFS Camp! 🇧🇪

🦋 @expede.wtf

🦣 @expede@octodon.social

📧 hello@brooklynzelenka.com

✍️ notes.brooklynzelenka.com