A Yarn of Randomness

The CU Randomness Beacon and the Twine protocol

Jasper Palfree - University of Colorado; NIST

Presented at the NIST Crypto Reading Club
2024-02-21

C
U
R
B
y

Colorado
University
R
B
y

Gautam Kavuri
CU/NIST

Jasper Palfree
CU/NIST

Krister Shalm
NIST

C
U
Randomness
Beacon
y

Source of public randomness

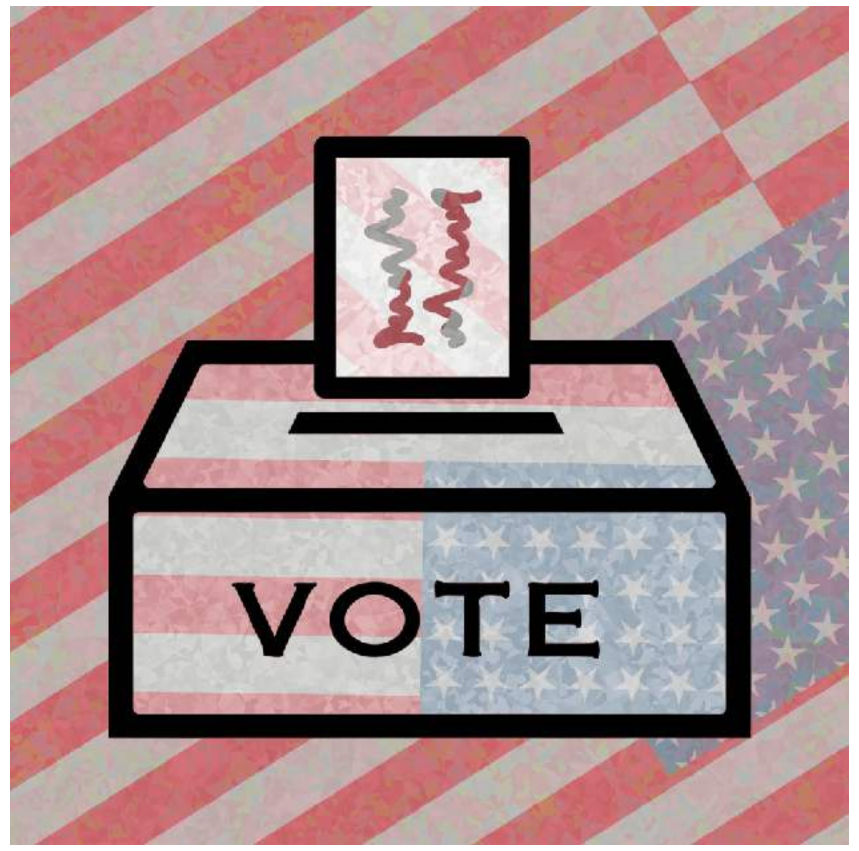
Emits bits at regular intervals (usually)

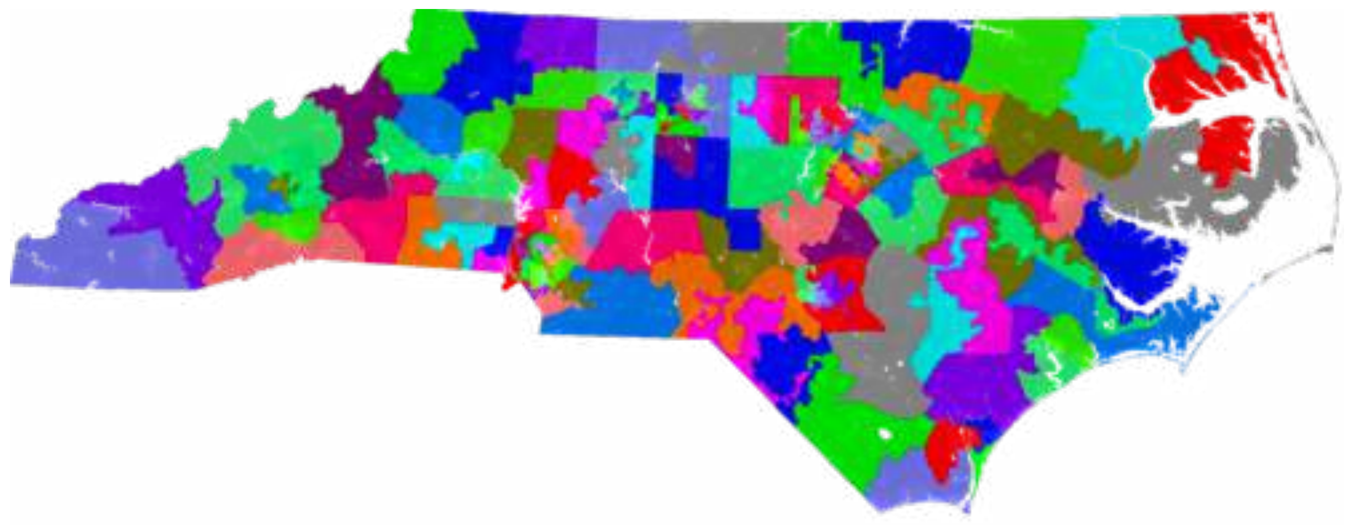
Can be verified to be unbiased and unpredictable

Maintains an immutable record of past values

C
U
R
B
y

Useful for enabling unbiased choices needed in high-stakes decisions





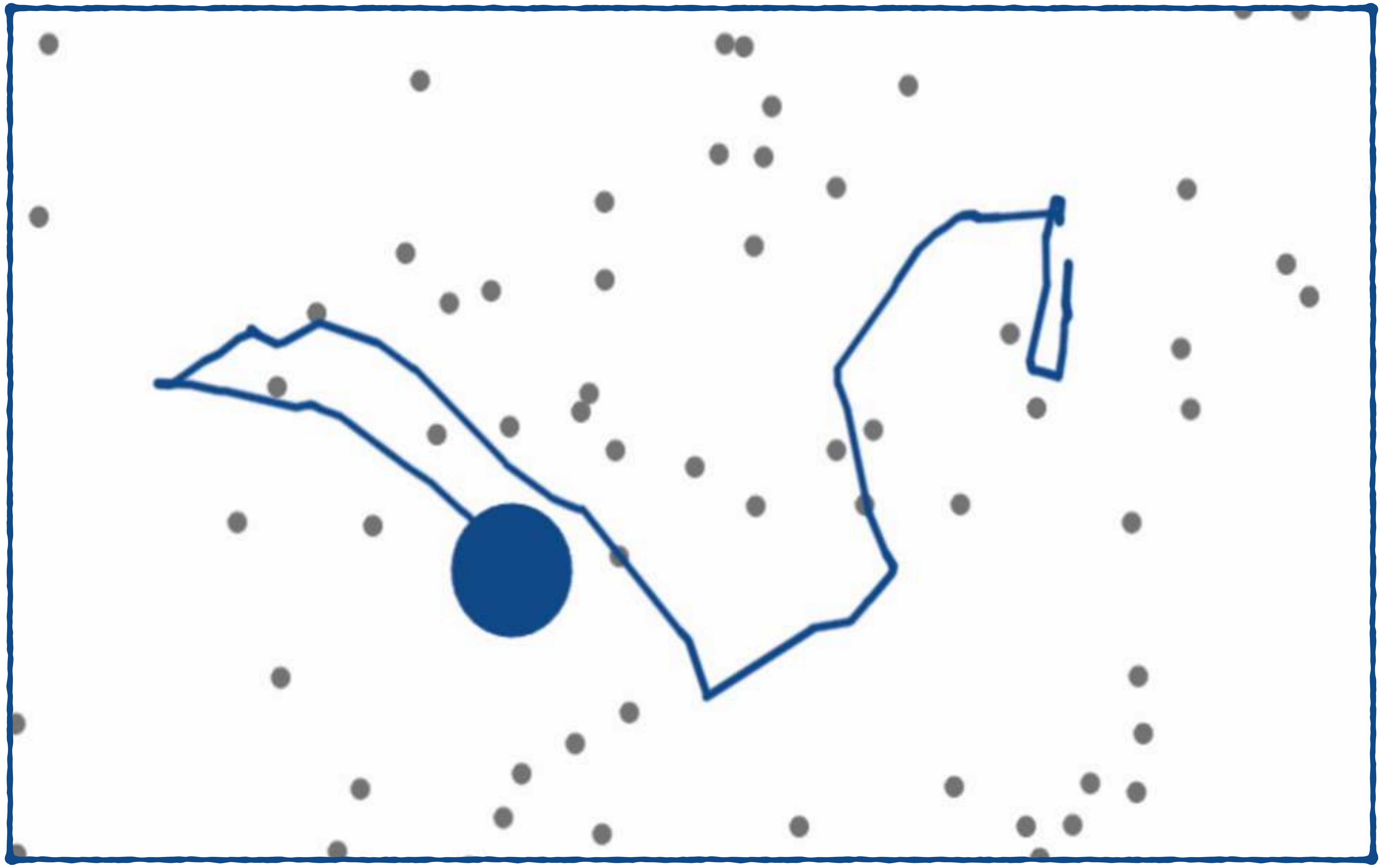
Why build another beacon?

There are already randomness beacons

- NIST Randomness Beacon Gaithersburg
- Singapore Randomness Beacon (in development)
- Random UChile
- DRAND (Cloudflare)

2003

2003

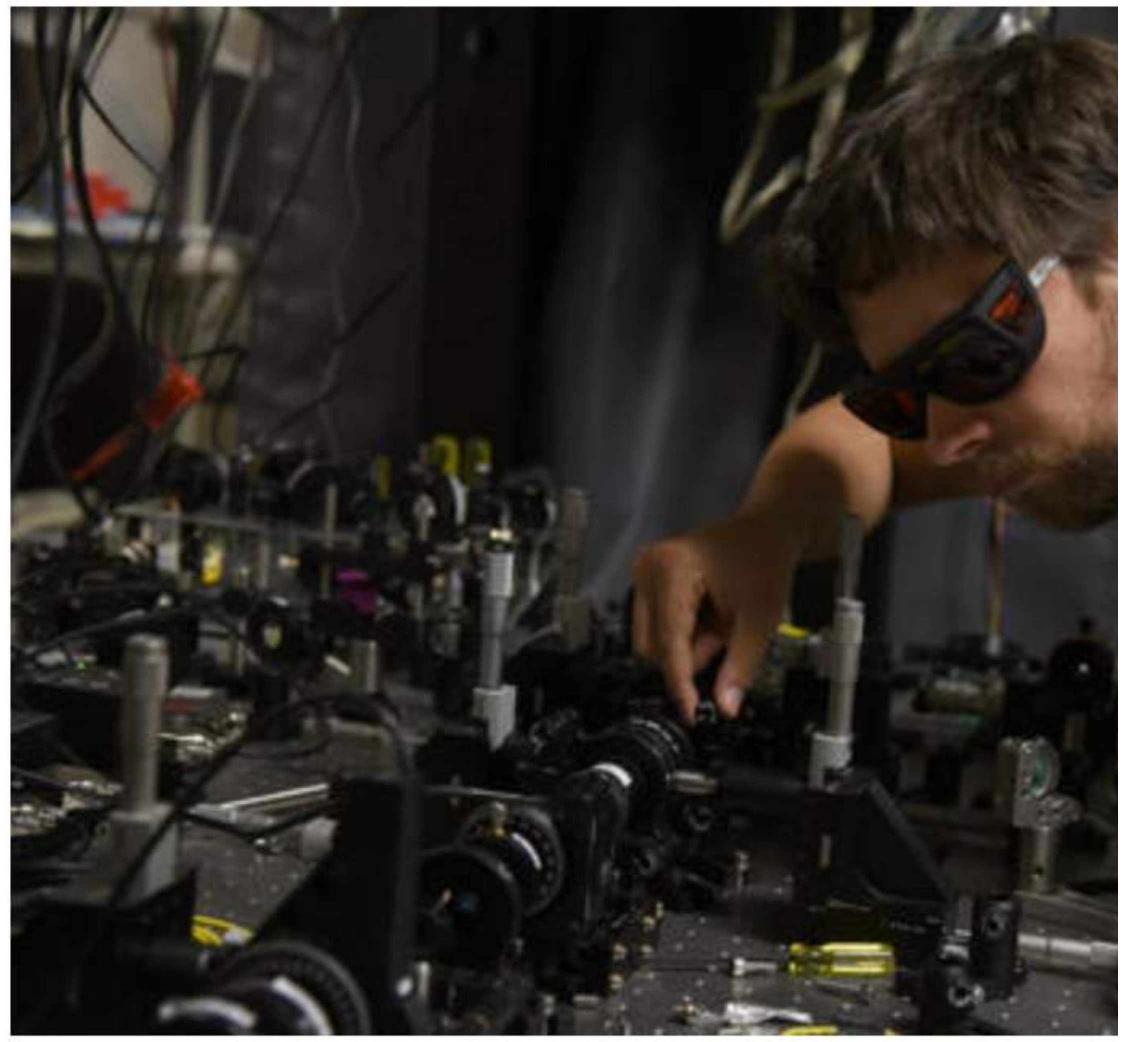


The Bell Experiment confirms that nature, as described by quantum mechanics, behaves in a fundamentally non-deterministic way.

NIST Team Proves 'Spooky Action at a Distance' is Really Real

November 10, 2015

BOULDER, Colo.—Einstein was wrong about at least one thing: There are, in fact, "spooky actions at a distance," as now proven by researchers at the National Institute of Standards and Technology (NIST).

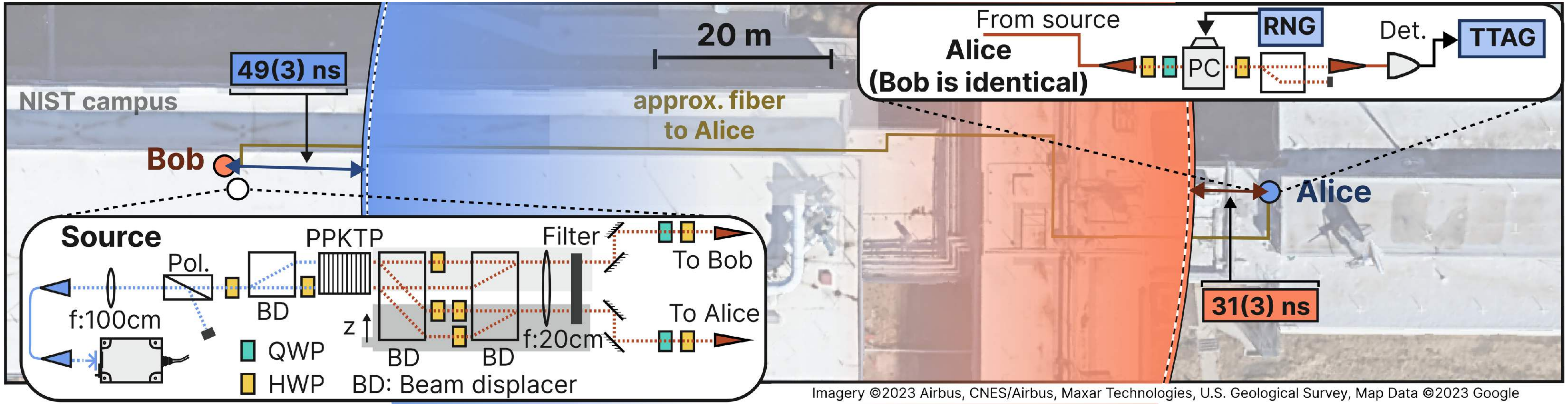


NIST physicist Krister Shalm with the photon source used in strongly supported a key prediction of quantum mechanics: spooky actions at a distance.
Credit: J. Burrus/NIST

Einstein used that term to refer to quantum mechanics, which describes the curious behavior of the smallest particles of matter and light. He was referring, specifically, to entanglement, the idea that two physically separated particles can have correlated properties, with values that are uncertain until they are measured. Einstein was dubious, and until now, researchers have been unable to support it with near-total confidence.

As described in a paper posted online and published in *Physical Review Letters* (PRL),* researchers from NIST and several other

Bell Experiment



Imagery ©2023 Airbus, CNES/Airbus, Maxar Technologies, U.S. Geological Survey, Map Data ©2023 Google

This is great for generating truly random numbers

DIRNG (Device-Independent Random Number Generation) uses Bell Experiment data to extract entropy from that non-deterministic part of nature.

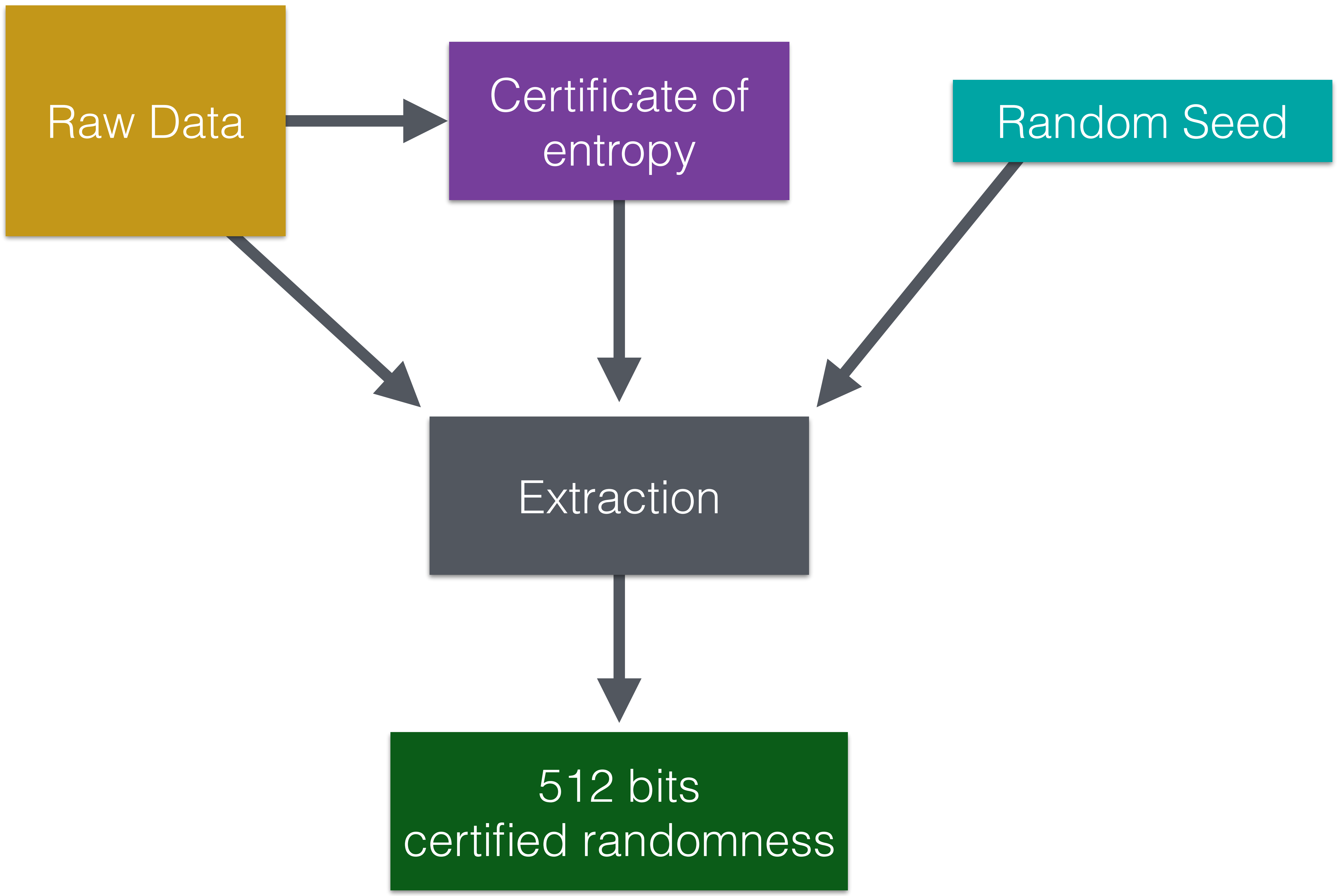
Experimental Low-Latency Device-Independent Quantum Randomness

Yanbao Zhang,[1,*] Lynden K. Shalm,[2,*] Joshua C. Bienfang,[3] Martin J. Stevens,[2] Michael D. Mazurek,[2] Sae Woo Nam,[2] Carlos Abellán,[4,†] Waldimar Amaya,[4,†] Morgan W. Mitchell,[4,5] Honghao Fu,[6] Carl A. Miller,[6,3] Alan Mink,[3,7] and Emanuel Knill[2,8]

[1] *NTT Basic Research Laboratories and NTT Research Center for Theoretical Quantum Physics, NTT Corporation, 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198, Japan*
[2] *National Institute of Standards and Technology, Boulder, Colorado 80305, USA*
[3] *National Institute of Standards and Technology, Gaithersburg, MD 20899, USA*
[4] *ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain*
[5] *ICREA-Institució Catalana de Recerca i Estudis Avançats, 08010 Barcelona, Spain*
[6] *Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742, USA*
[7] *Theiss Research, La Jolla, CA 92037, USA*
[8] *Center for Theory of Quantum Matter, University of Colorado, Boulder, Colorado 80309, USA*

Applications of randomness such as private key generation and public randomness beacons require small blocks of certified random bits on demand. Device-independent quantum random number generators can produce such random bits, but existing quantum-proof protocols and loophole-free implementations suffer from high latency, requiring many hours to produce any random bits. We demonstrate device-independent quantum randomness generation from a loophole-free Bell test with a more efficient quantum-proof protocol, obtaining multiple blocks of 512 random bits with an average experiment time of less than 5 min per block and with a certified error bounded by $2^{-64} \approx 5.42 \times 10^{-20}$.

DIRNG (Device-Independent Random Number Generation)



Benefits

With other RNG methods it's very difficult to verify the process is truly random.

With DIRNG the output is certified to have a certain amount of entropy, and that entropy must have come from a non-deterministic process.

...regardless of the experimental setup.

Building a randomness beacon

NIST wrote instructions

Draft NISTIR 8213

A Reference for Randomness Beacons

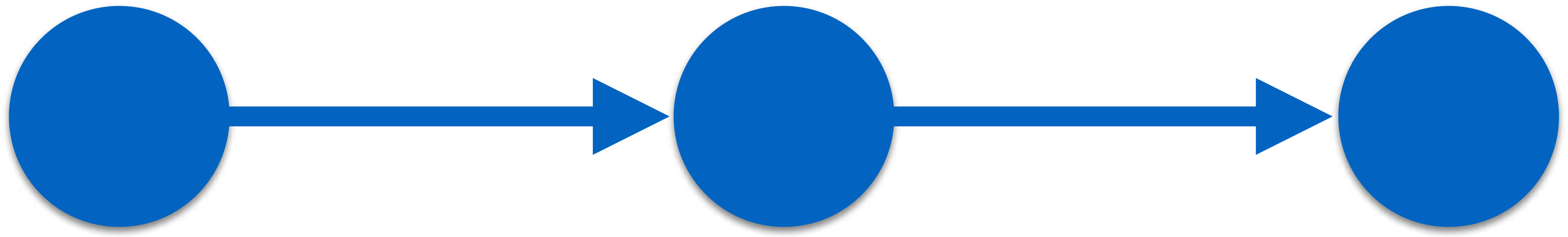
Format and Protocol Version 2

John Kelsey
Luís T. A. N. Brandão
René Peralta
Harold Booth

Building a randomness beacon

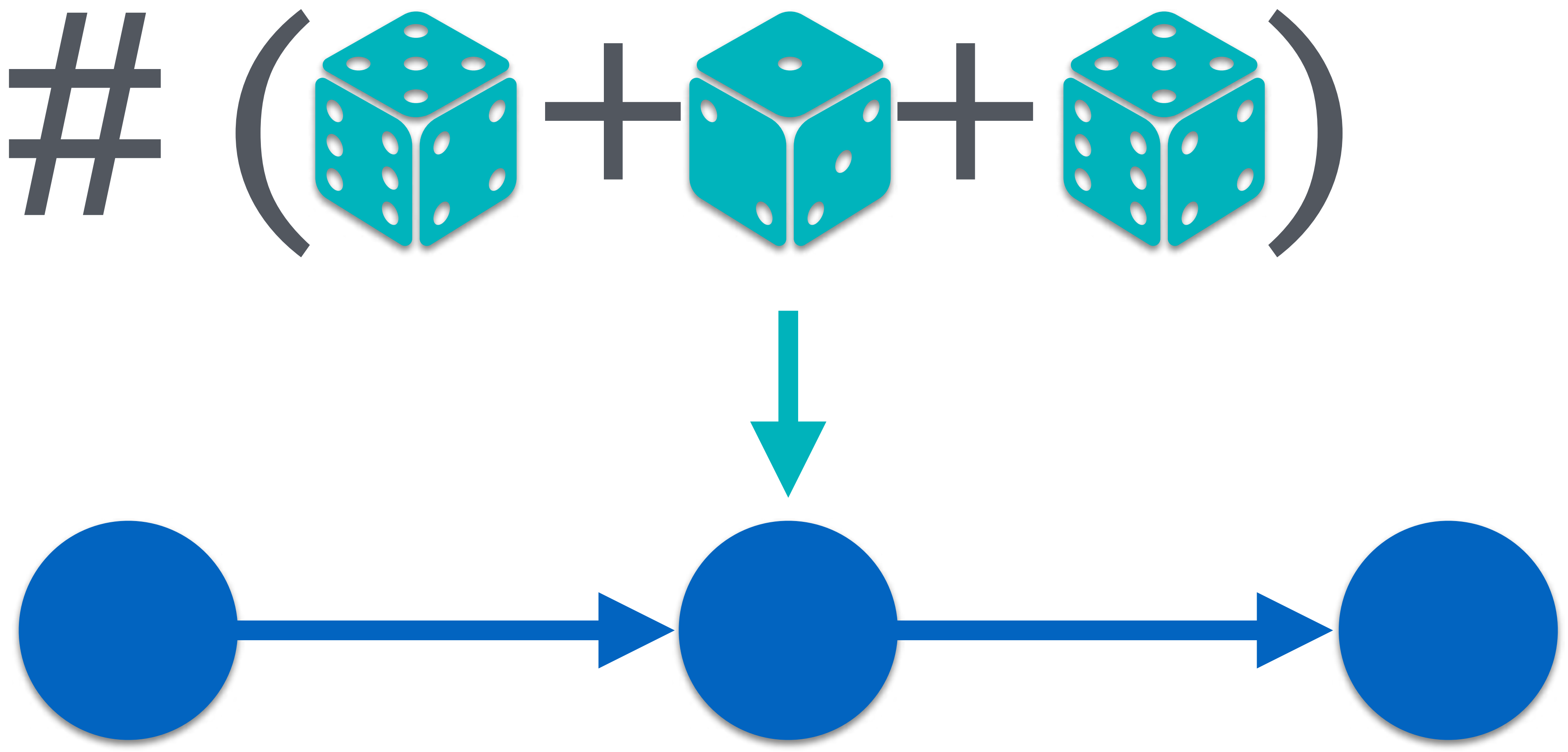
In a nutshell...

A hash-chain of digitally signed pulses



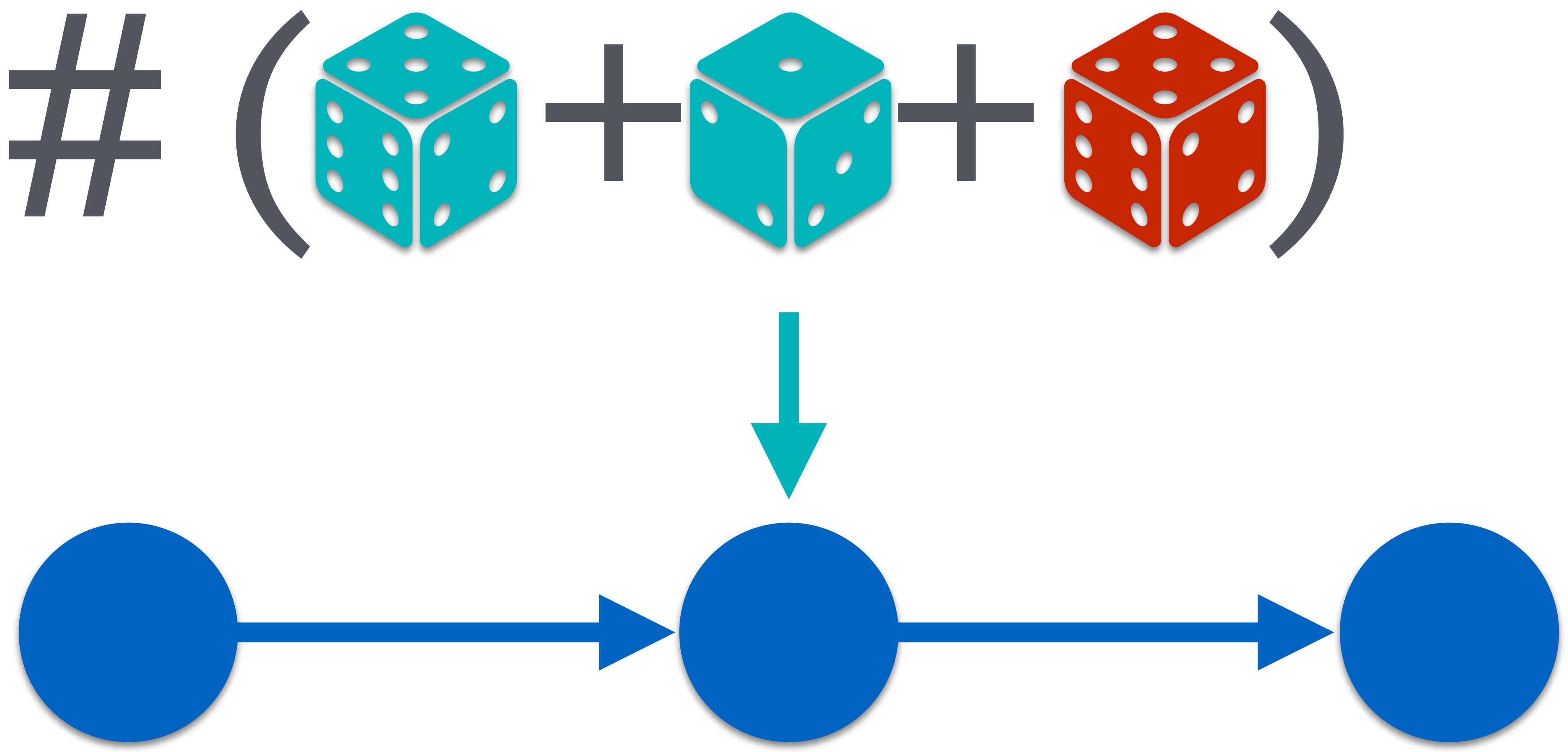
Building a randomness beacon

In a nutshell...



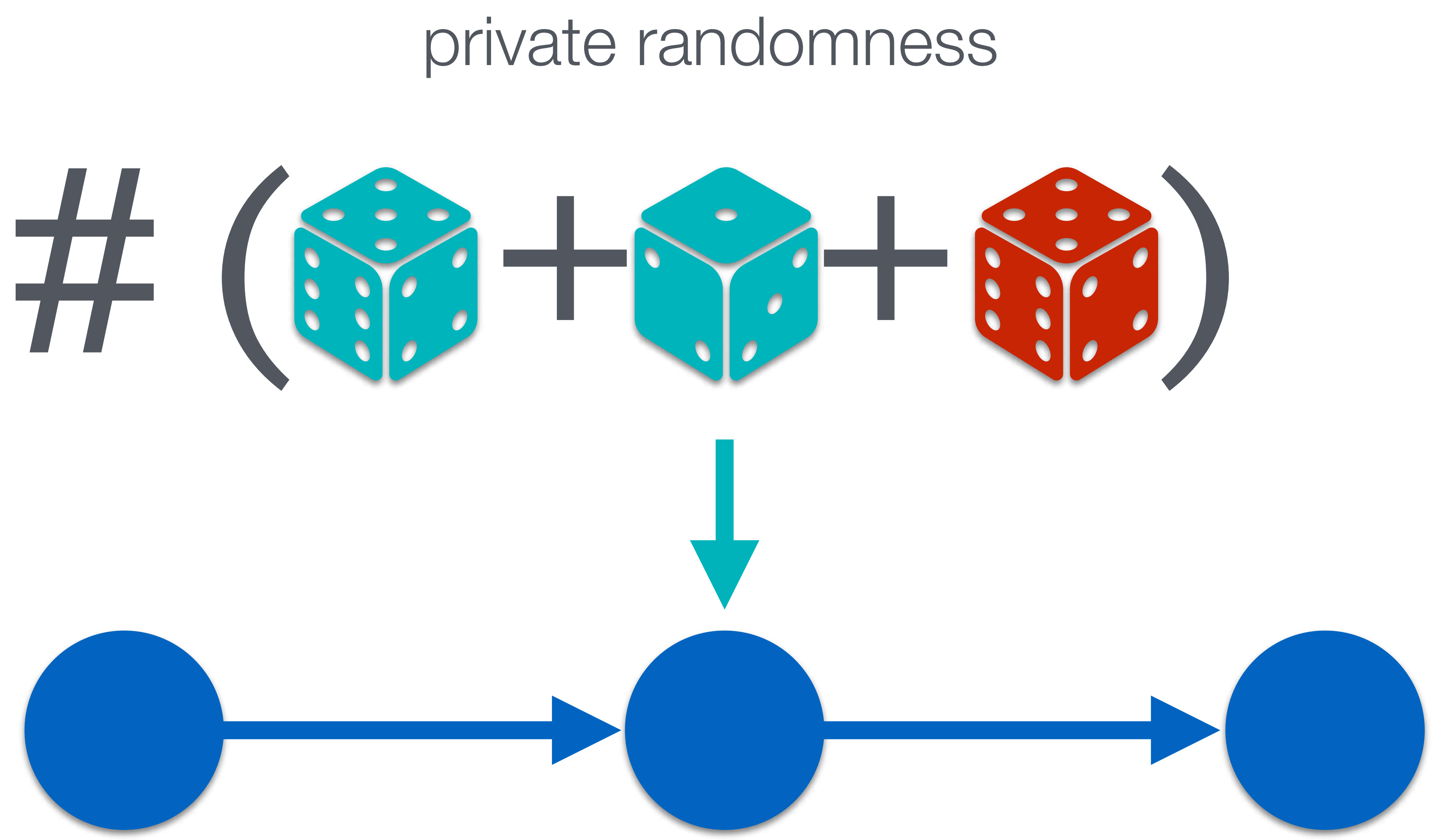
Building a randomness beacon

In a nutshell...



Building a randomness beacon

In a nutshell...



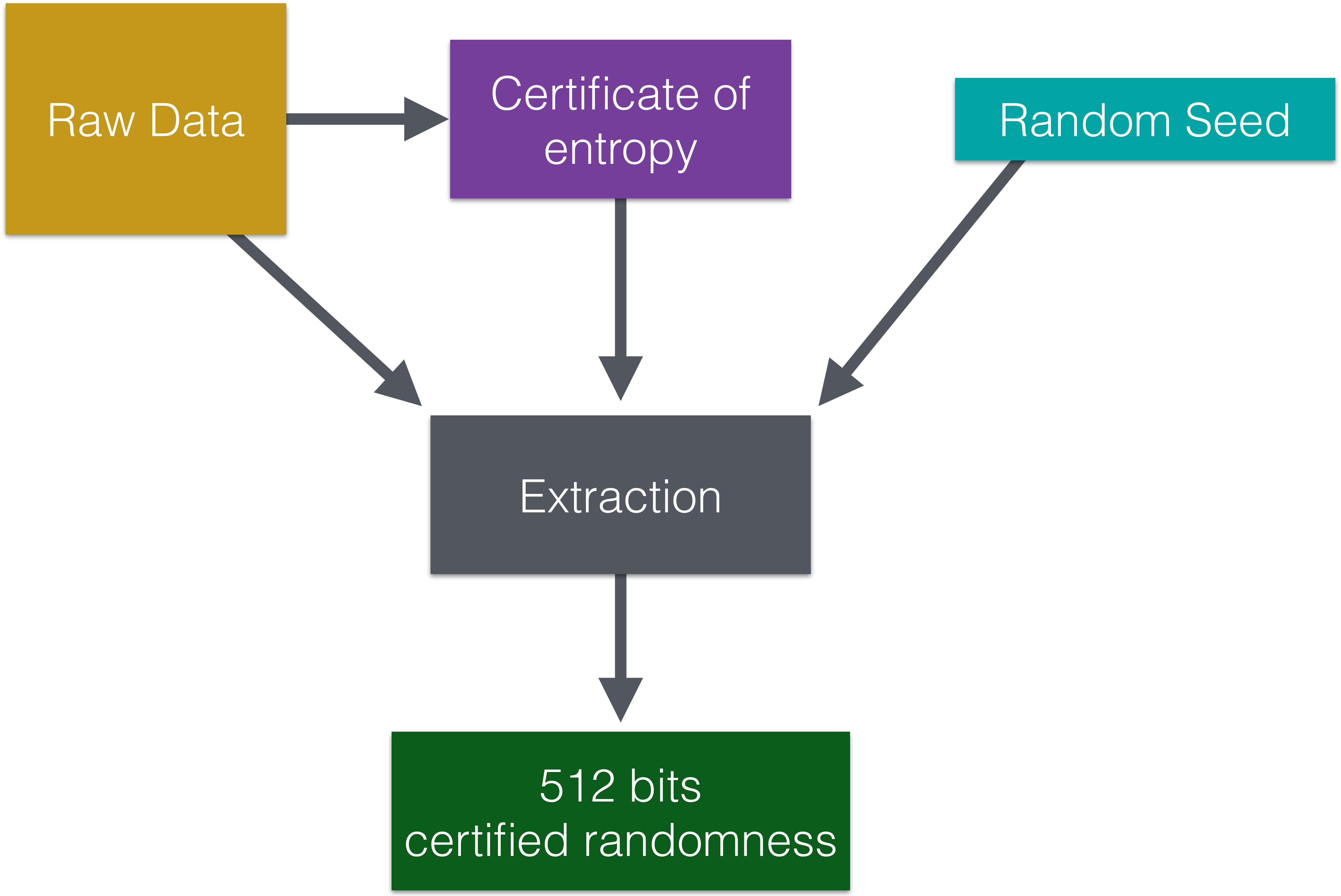
Building a randomness beacon

Challenge 2:

DIRNG extracts randomness from raw bell data using a random seed which influences the output randomness.

Need a way to prove that the seed was chosen by a neutral party.

Need a way to prove that we pre-committed to using a value that we didn't know at the time we committed to it.

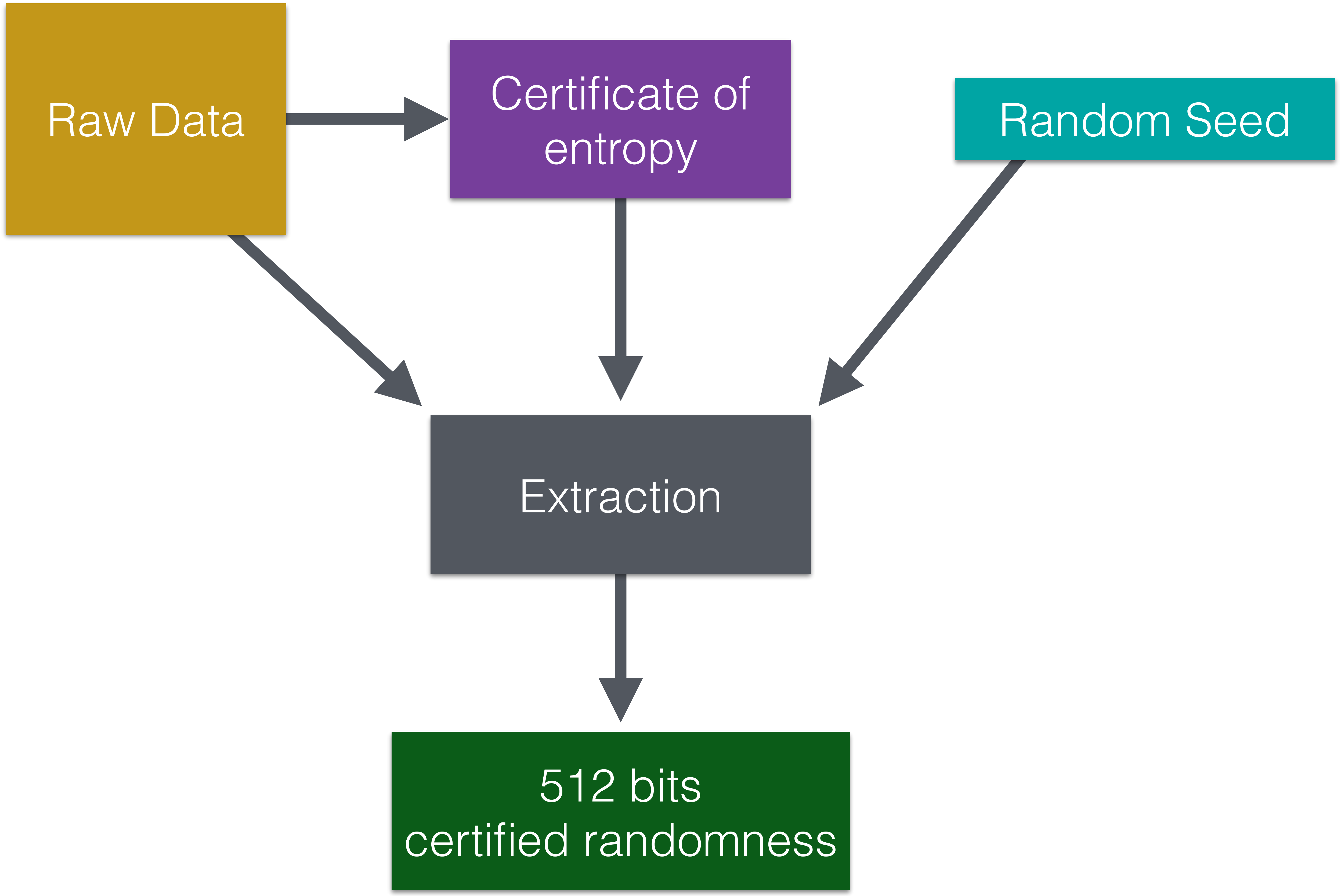


Building a randomness beacon

One protocol to rule them all?

The Twine Protocol

Three separate processes = Three separate chains



One protocol to rule them all?

The Twine Protocol

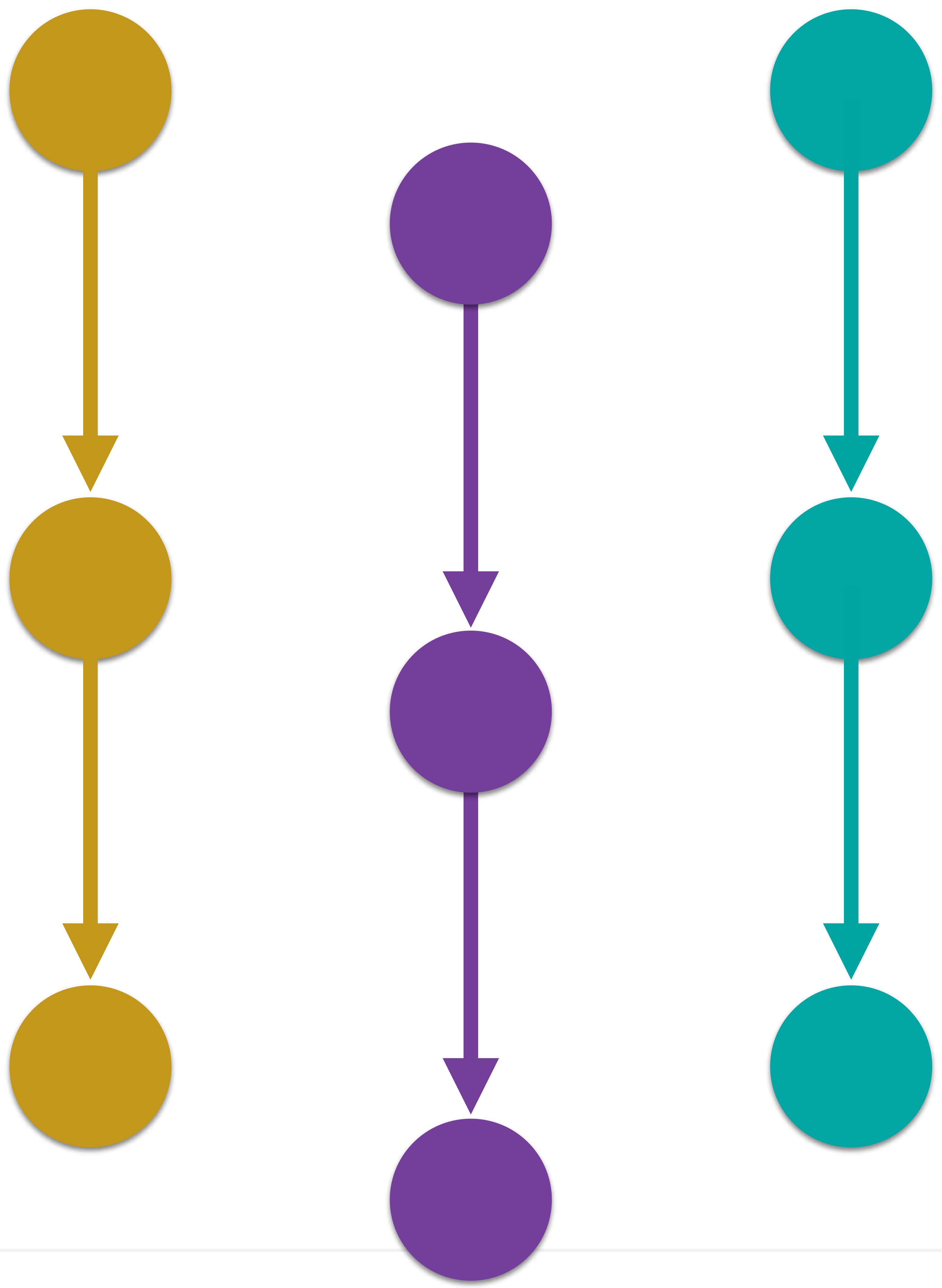
Three separate processes = Three separate chains

Pulses are generic containers.

Each chain can have its own sub-protocol
for its own use-case.

Twine pulse

```
{
  ...
  specification: "twine/1.0.x/nist-rng/1.0.x"
  payload: { arbitrary data }
}
```



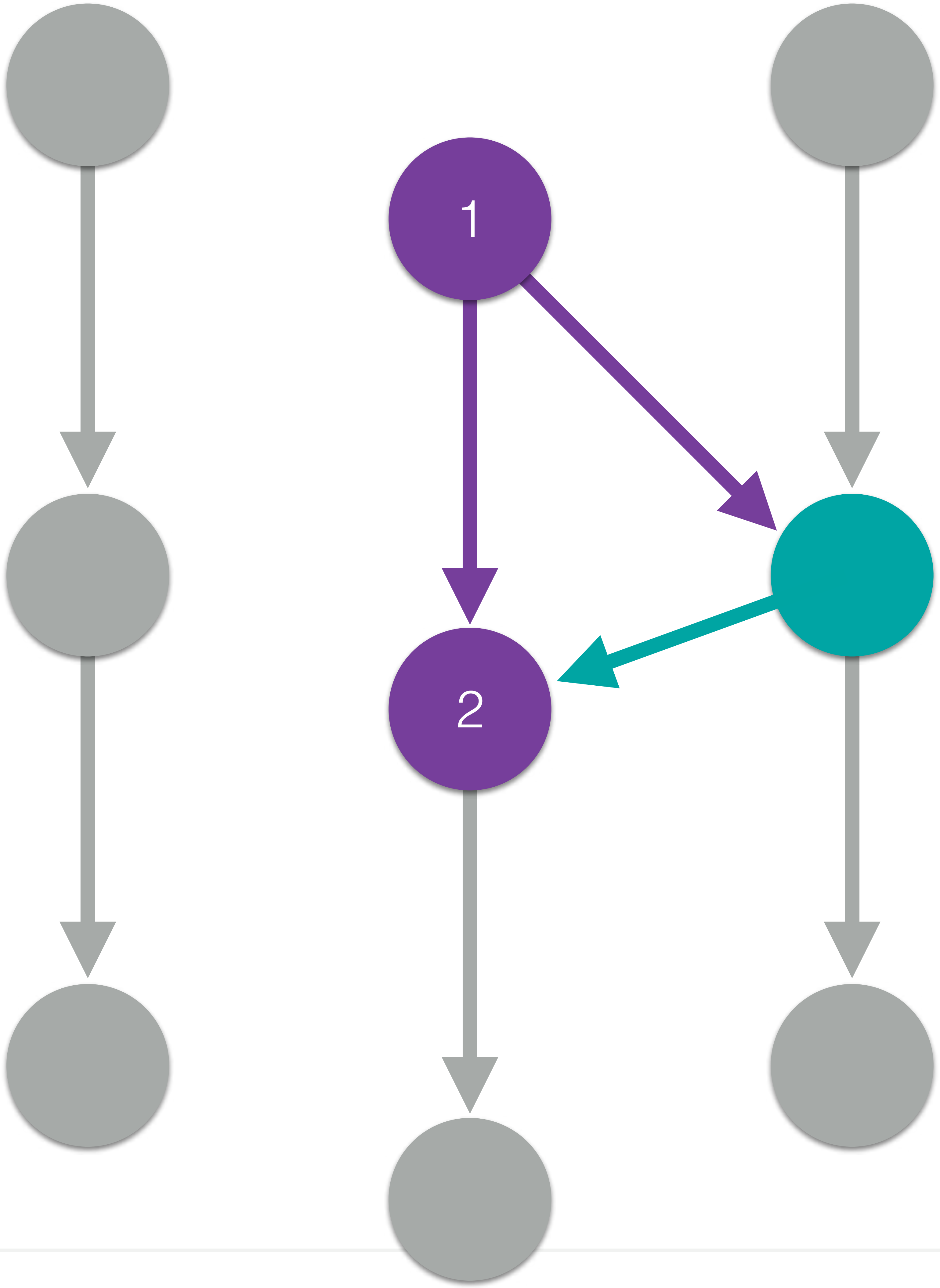
Cross-chain hash linking

The Twine Protocol

Pulses include hashes of pulses on other chains

Serves two purposes:

1. Time bound on the output hash. Blue pulse hash can not be known until after purple pulse is published and vice versa.



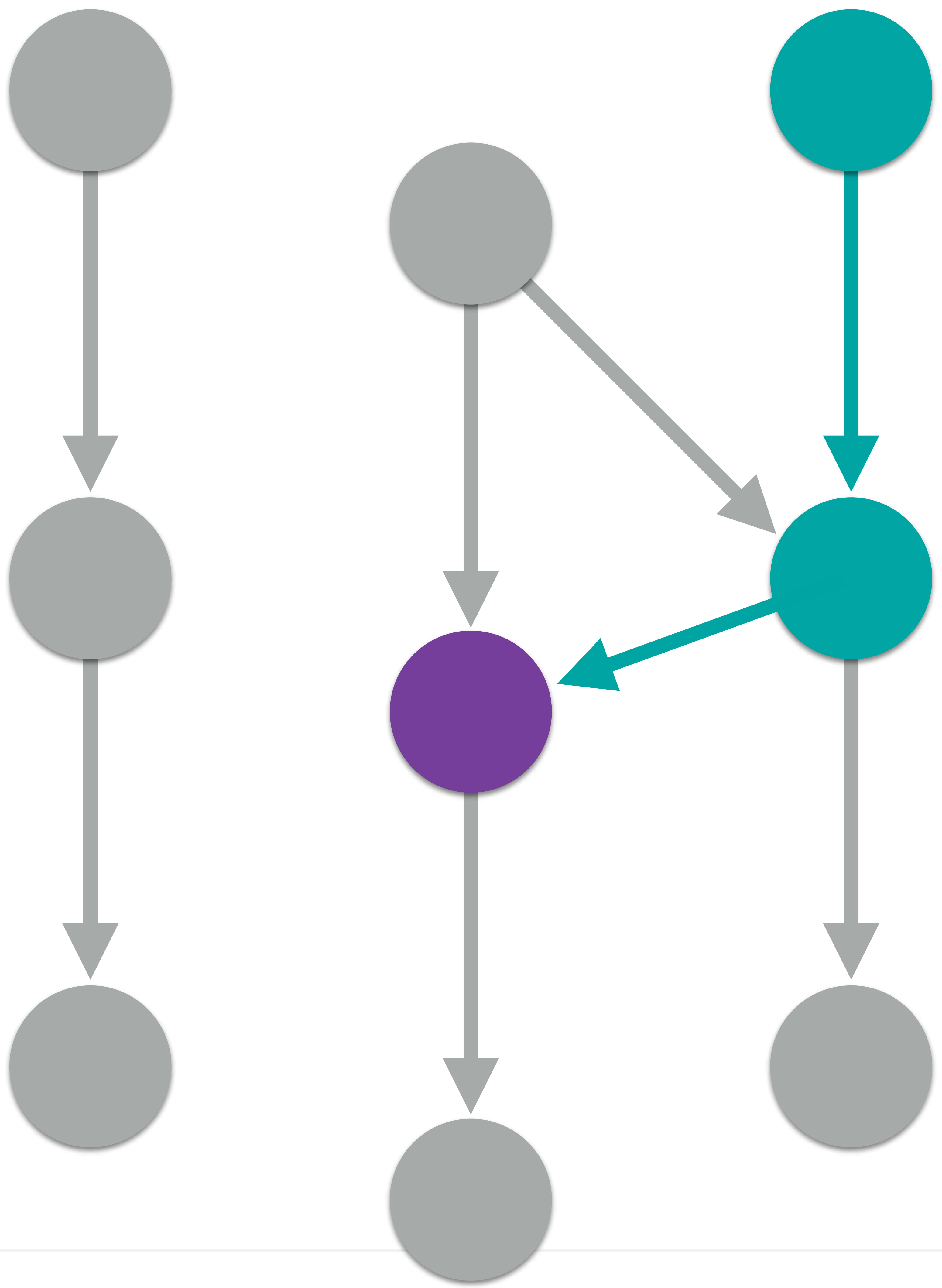
Cross-chain hash linking

The Twine Protocol

Pulses include hashes of pulses on other chains

Serves two purposes:

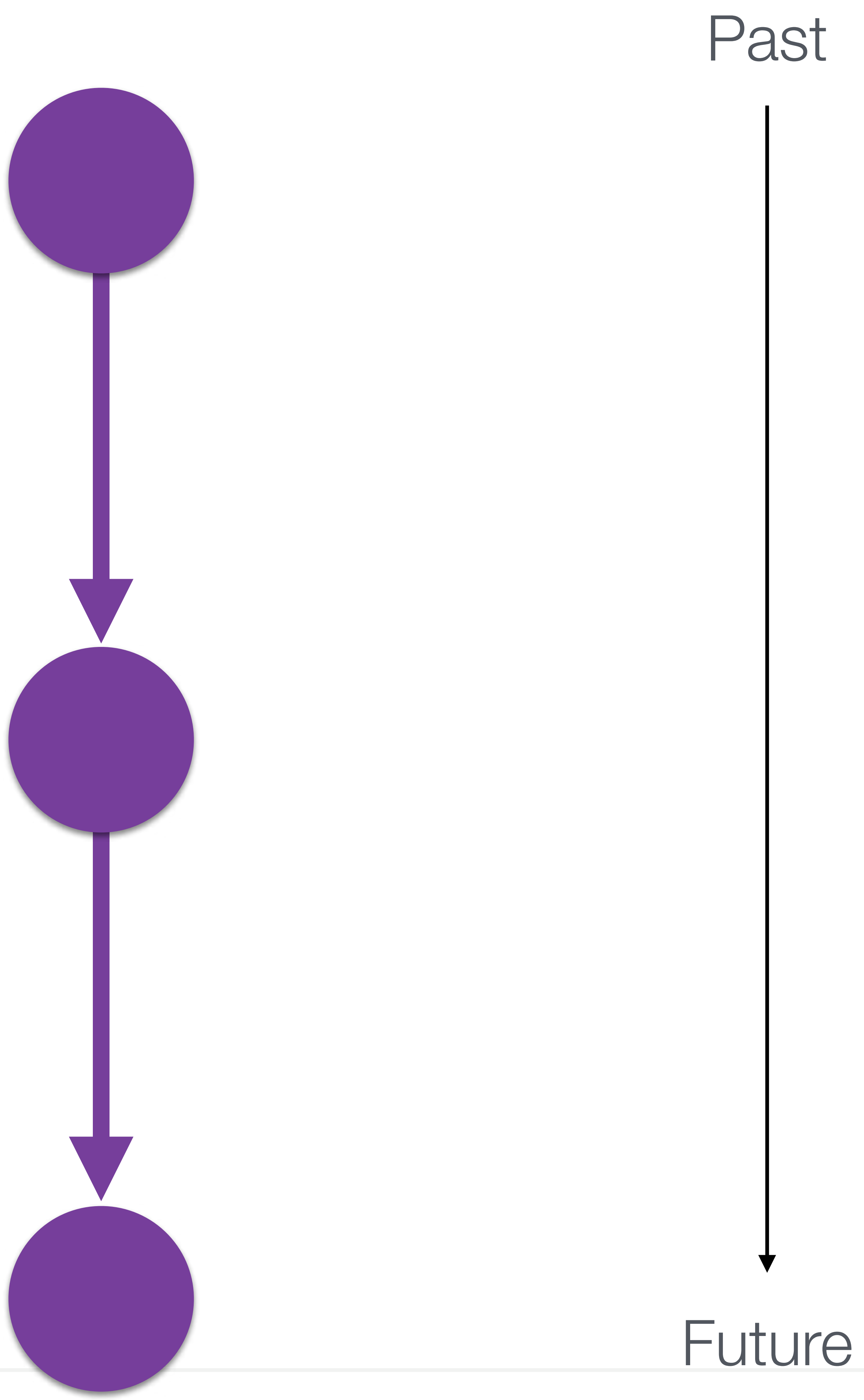
1. Time bound on the output hash. Blue pulse hash can not be known until after purple pulse is published and vice versa.
2. Preserves the history of the external chain. Blue chain's past pulses cannot be rewritten without purple chain knowing.



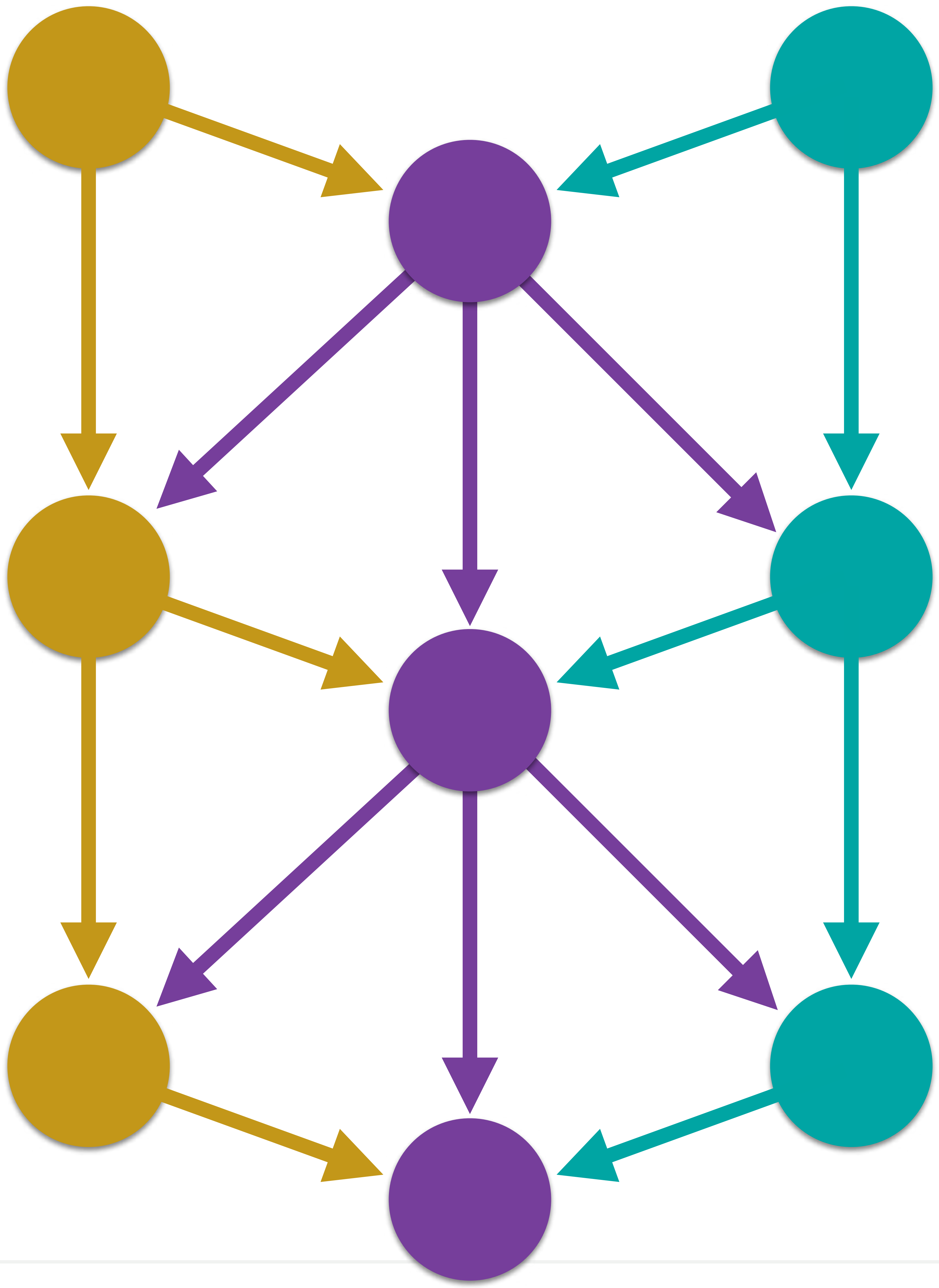
Stitching pulses of different chains together using hash-linking provides both tight time bounds and data integrity.

A stitch in Twine saves time

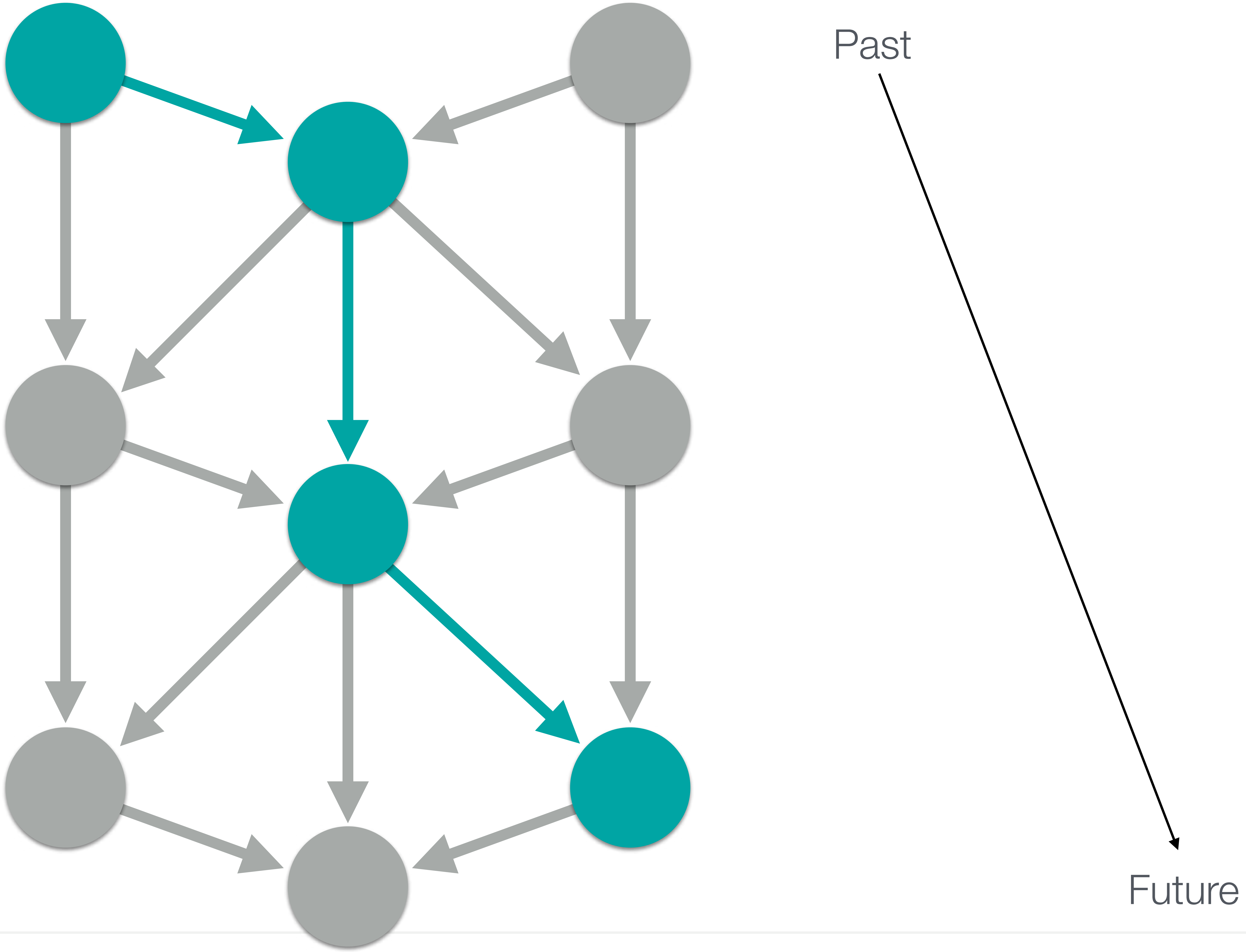
A different way to think about time-stamping



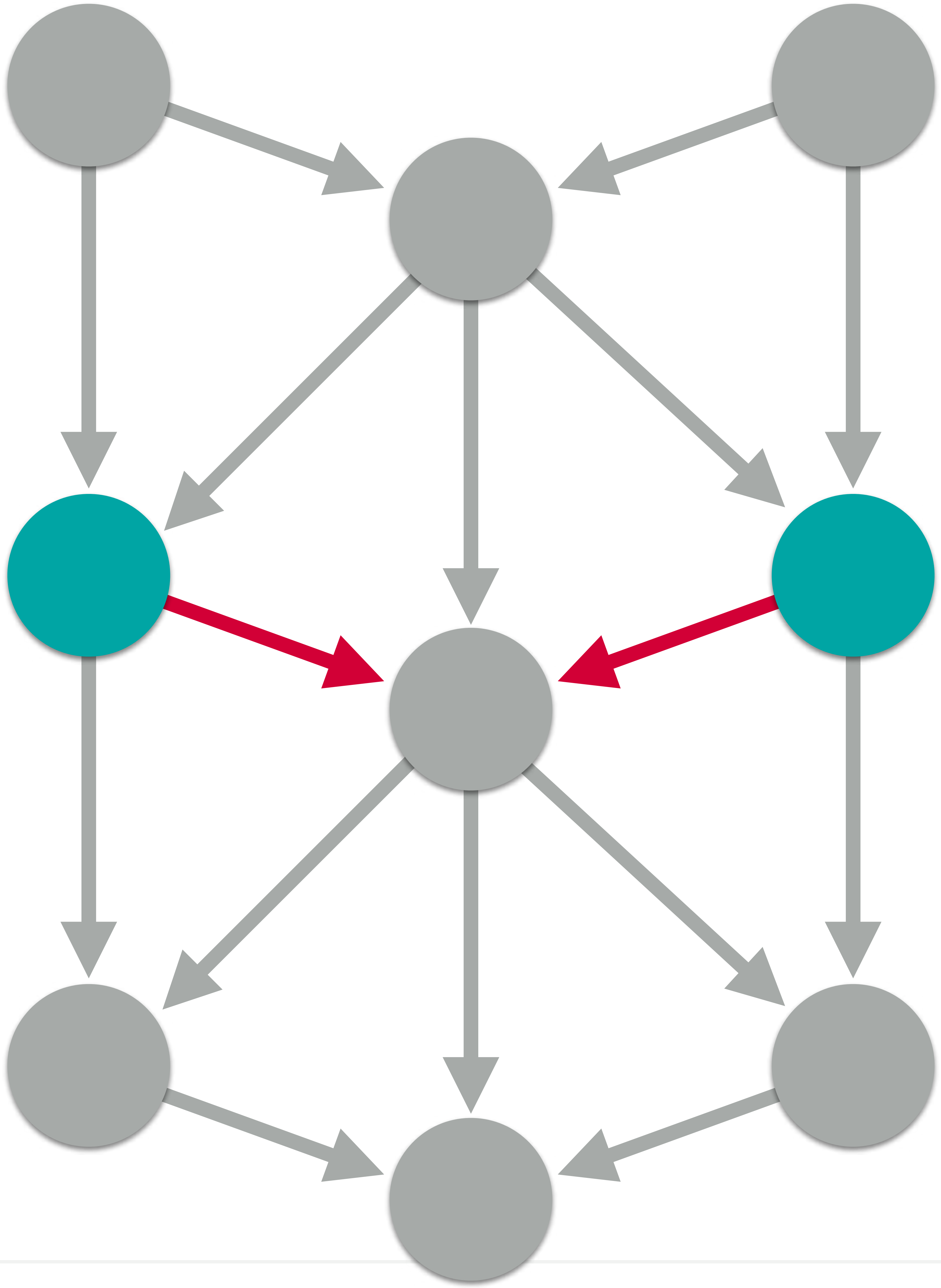
This creates a DAG (Directed Acyclic Graph)



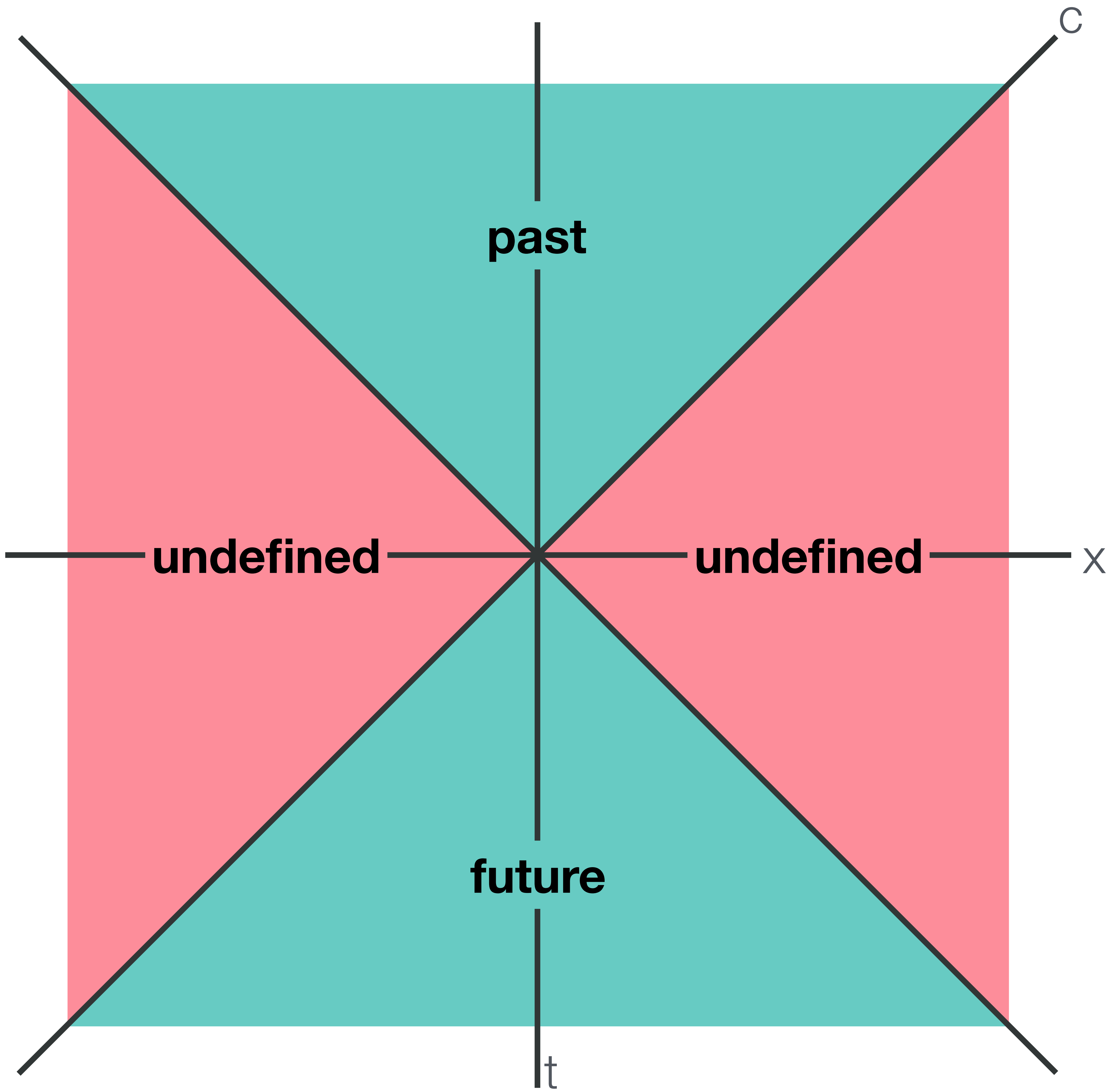
Pathfinding implies causal ordering



But in general a DAG only implies partial ordering

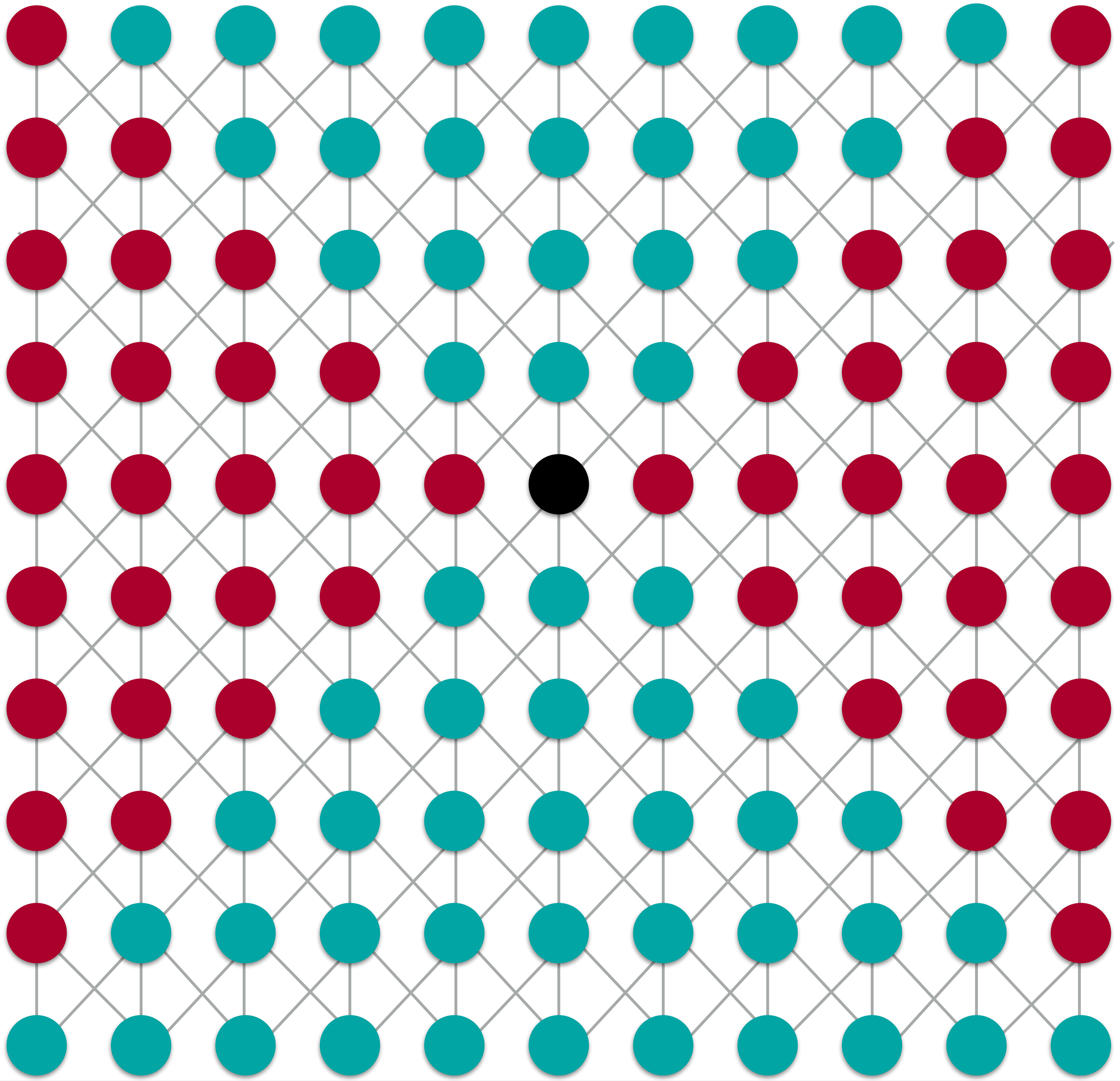


Compare it with spacetime



Stitching together chains creates a "hash-time"

Information propagates at the "hash-speed"



This is more faithfully reflects the true (relativistic) nature of time

It takes three to tangle

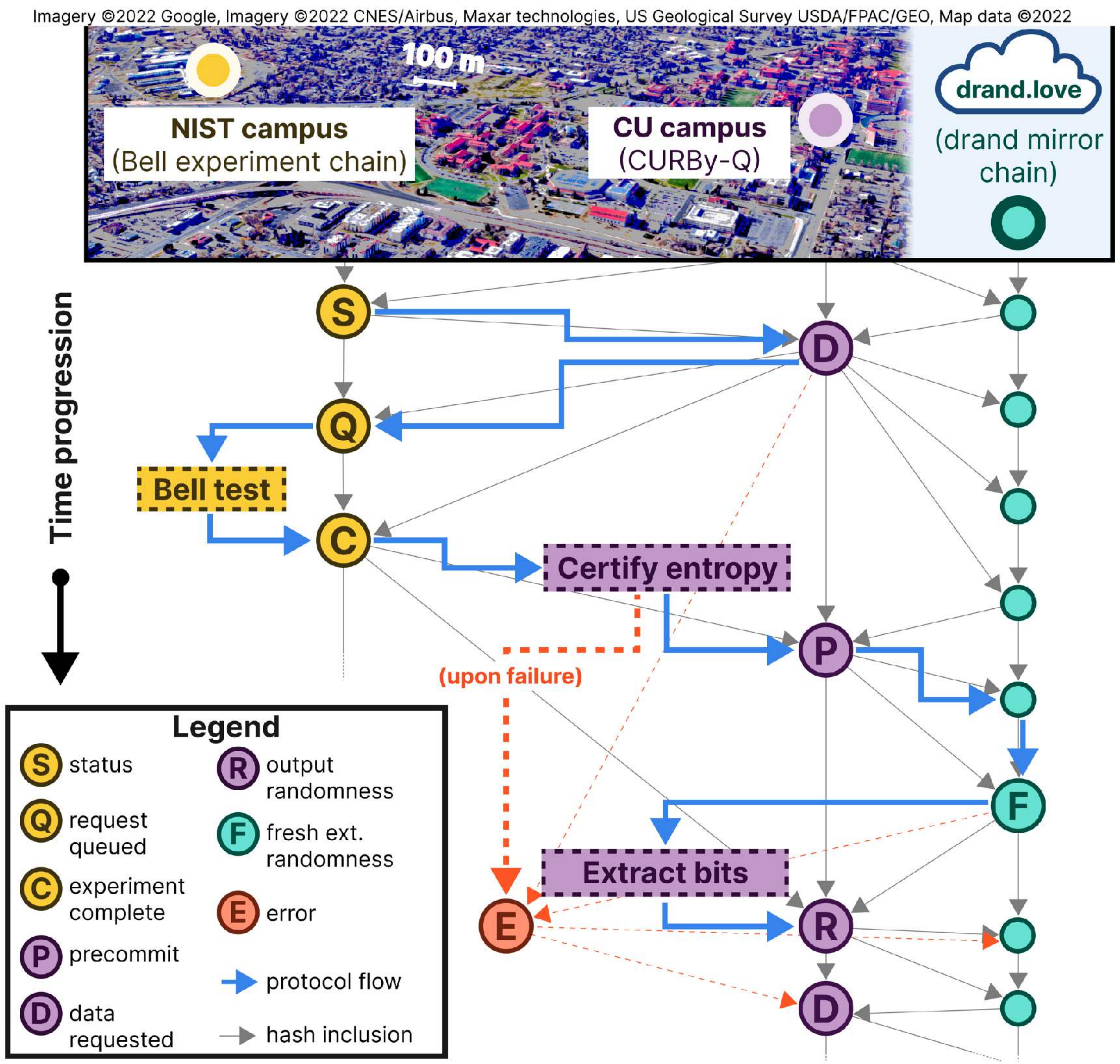
CU Chain carries out the DIRNG protocol
NIST Chain records experiment events
DRAND wrapper Chain acts as the seed chain

Communication happens through Twine.

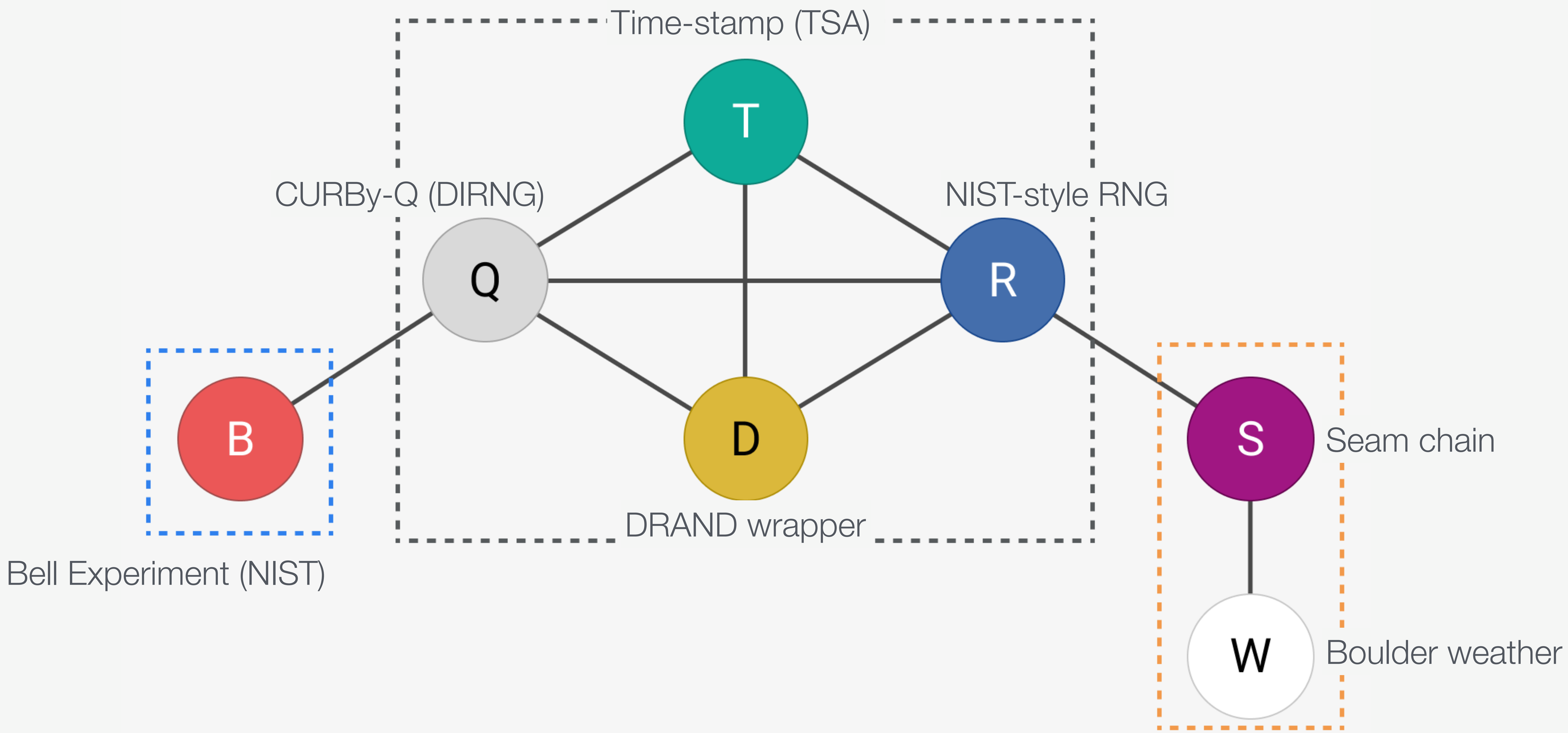
CU requests data. NIST provides data privately.

CU publishes precommitment pulse with hash of raw data, seed length, seed source (chain).

Once seed becomes available, randomness is extracted and published along with the raw data.



The CURBy Tangle



Live demo?

CURBy (CU Randomness Beacon)

What is CURBy?

CURBy is an acronym for the **CU R**andomness **B**eacon (why? because we can!). CURBy is a public service free for anyone to use. Its goal is to provide a new way to generate random values extracted from the quantum vacuum in a way that can be verified by anyone.

You can browse the past and present random values below:

CURBy | CURBy-Quantum

Viewing Pulse ID: b...ayzkvvlq
Mixing in with 5 other chains

c9c368845d6a508e8f9a7e2a9414d1ccb50c53abfe57a9a0f089154e2a52015f
a6baefeb4be8ba5296f66c1683bb7986061373621037 02425d7be18319556ab8

first << < 161507 > >> last

released: 14:38:00 on Thursday, February 15, 2024 pulse: 161507 / 161507

Why create a randomness beacon?

Often randomness is thought of as something you want kept hidden, such as when generating passwords or cryptographic keys. But broadcasting random values that everyone can agree on is incredibly useful for making decisions without human bias.

Consider the situation of deciding which ballot boxes should be audited in an election. Allowing humans, especially those in power, to make this decision leaves our democratic process more susceptible to corruption. Instead, guaranteeing that the decision is random and proving it to be outside of the control of any organization helps maintain the fairness of elections. This is what CURBy can do; prove that a random choice was unpredictable and outside of anyone's control.

What makes CURBy special?

Publicly agreed upon randomness is not a new idea. There have been many randomness beacons before CURBy, and CURBy's implementation has taken inspiration from its predecessors such as The League of Entropy's Distributed Randomness Beacon and especially The NIST Randomness Beacon.

What makes CURBy special is that it addresses two main pitfalls found in past beacons.

Challenge #1: Unpredictability vs. Randomness

The League of Entropy's beacon involves a network of computers that collectively generate pseudo-random values at regular intervals. If the nodes in the network are operated independently up to some threshold, the values are unpredictable, but they are still *deterministic;* that is, pre-determined from the start. This means that, while perhaps unlikely, it is possible for a large enough subset of nodes to coordinate and predict future values.

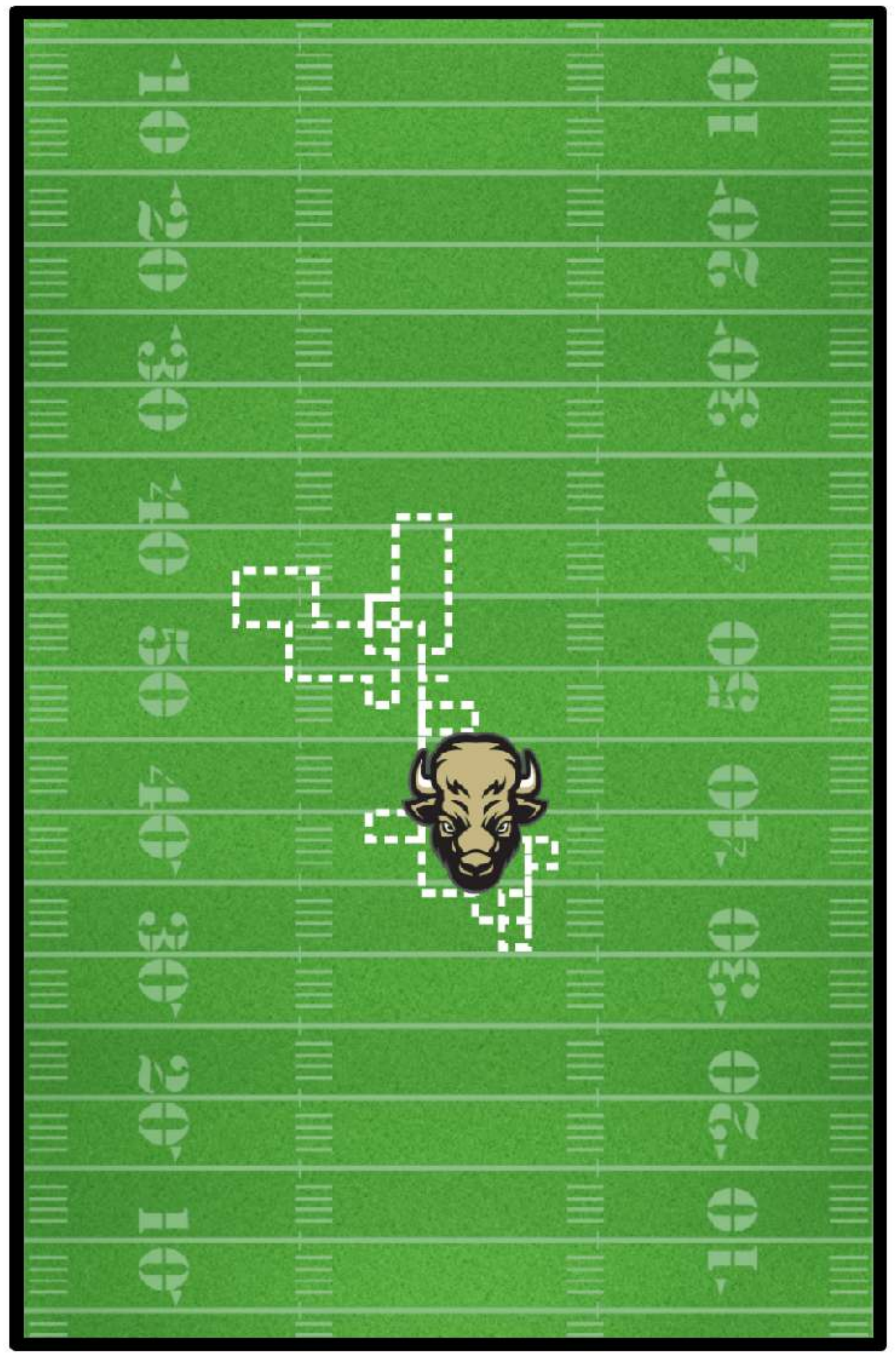
Challenge #2: Trust vs. Verifiability

Next randomness in 18 seconds

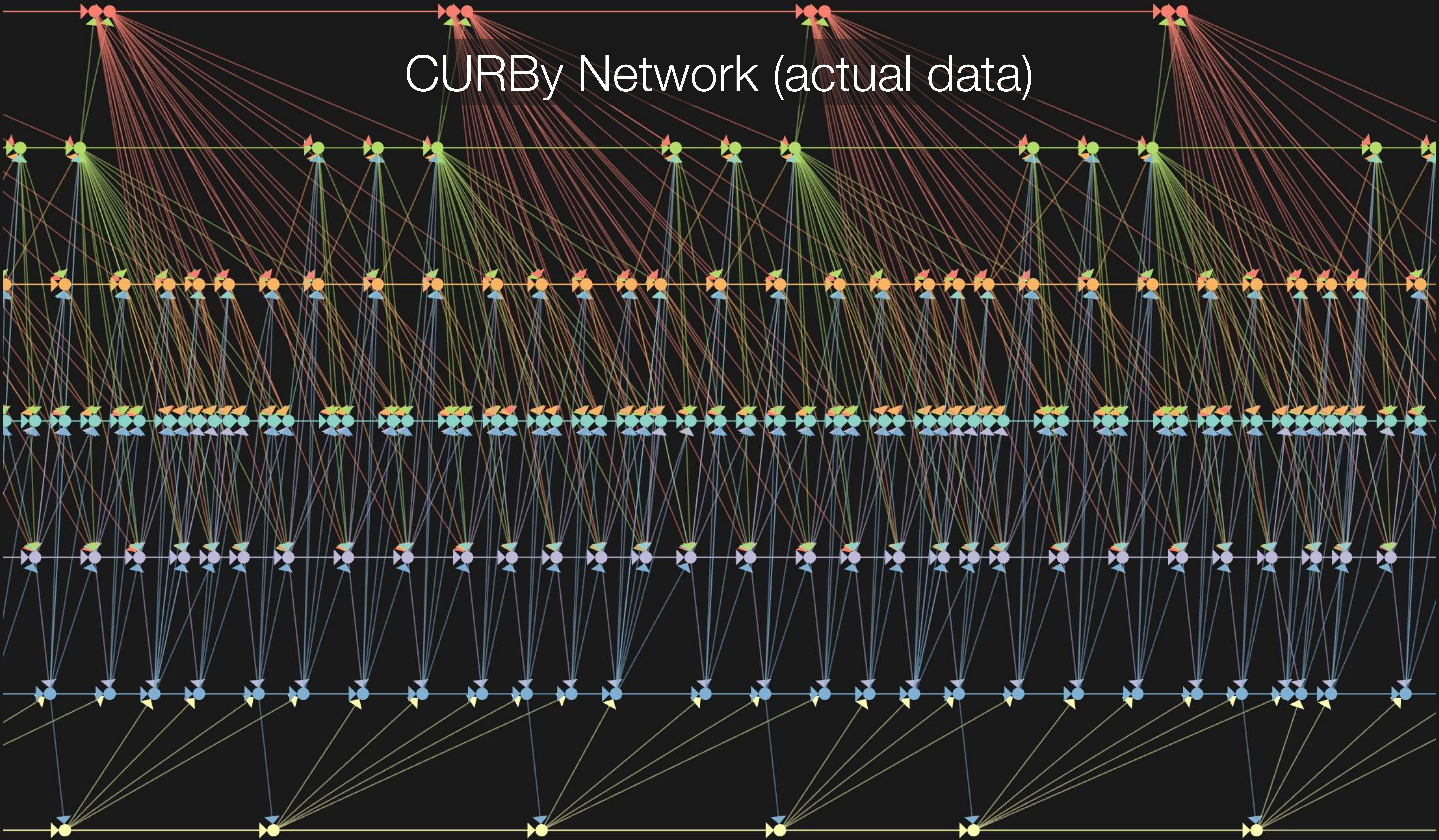
c9c368845d6a508e8f9a7e2a9414d1cc
b50c53abfe57a9a0f089154e2a52015f
a6baefeb4be8ba5296f66c1683bb7986
061373621037 02425d7be18319556ab8

Ralphie's Random Run

Here Ralphie, CU's mascot, is doing a random walk based on the latest random pulse.

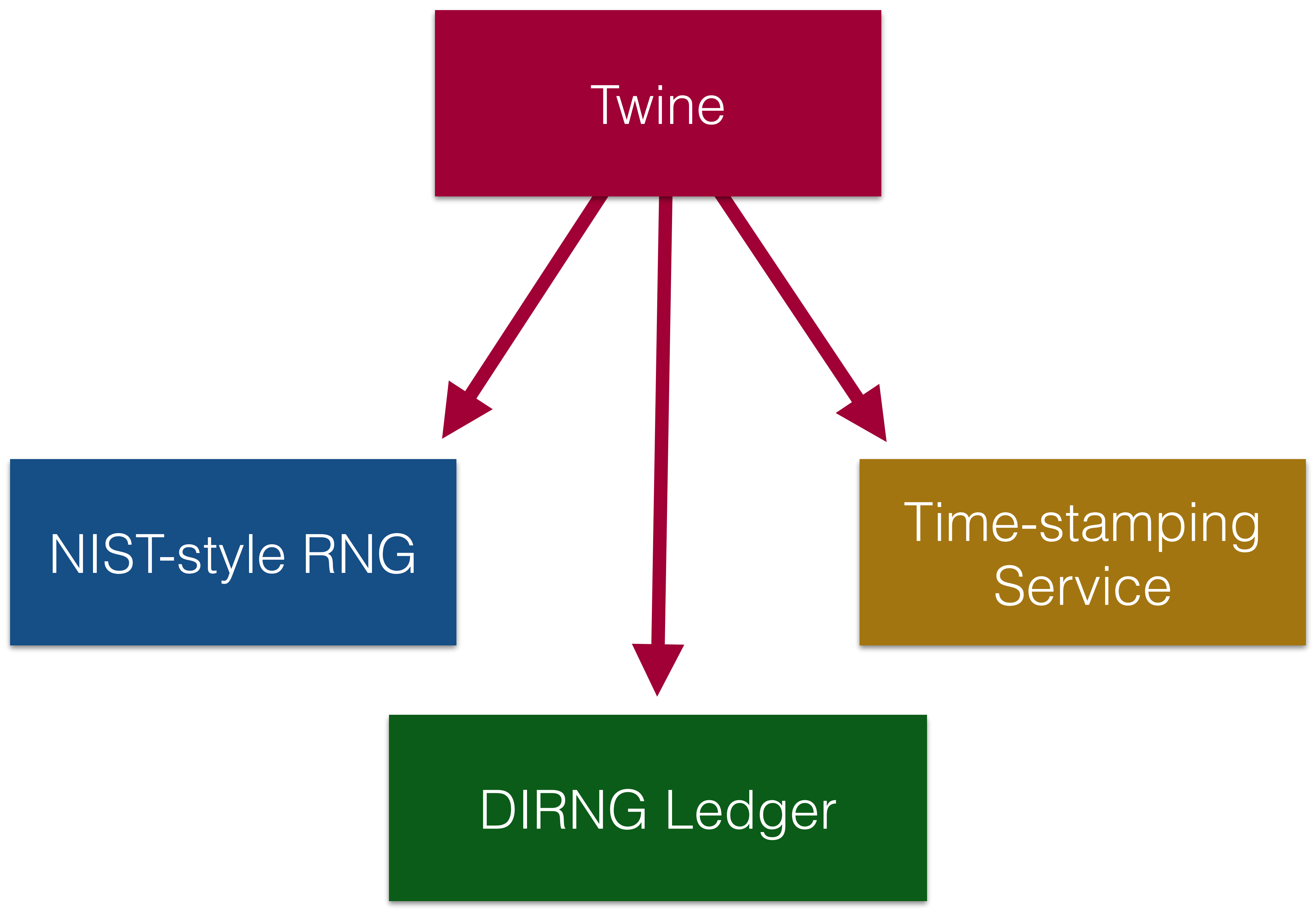


CURBy Network (actual data)



Features that make Twine flexible

- Generic payloads and sub-protocols.



Features that make Twine flexible

- Generic payloads and sub-protocols.
- Content addressing and CIDs (hashes are the identifier)

CID INFO

QmVWTzdzoK2znNPexzdFoZ7HGiTUCQG7Hh3xqy94czyJQC

base58btc - cidv0 - dag-pb - sha2-256~256~6A858049CB305292AA7084FAB18A5...

BASE - VERSION - CODEC - MULTIHASH

MULTIHASH

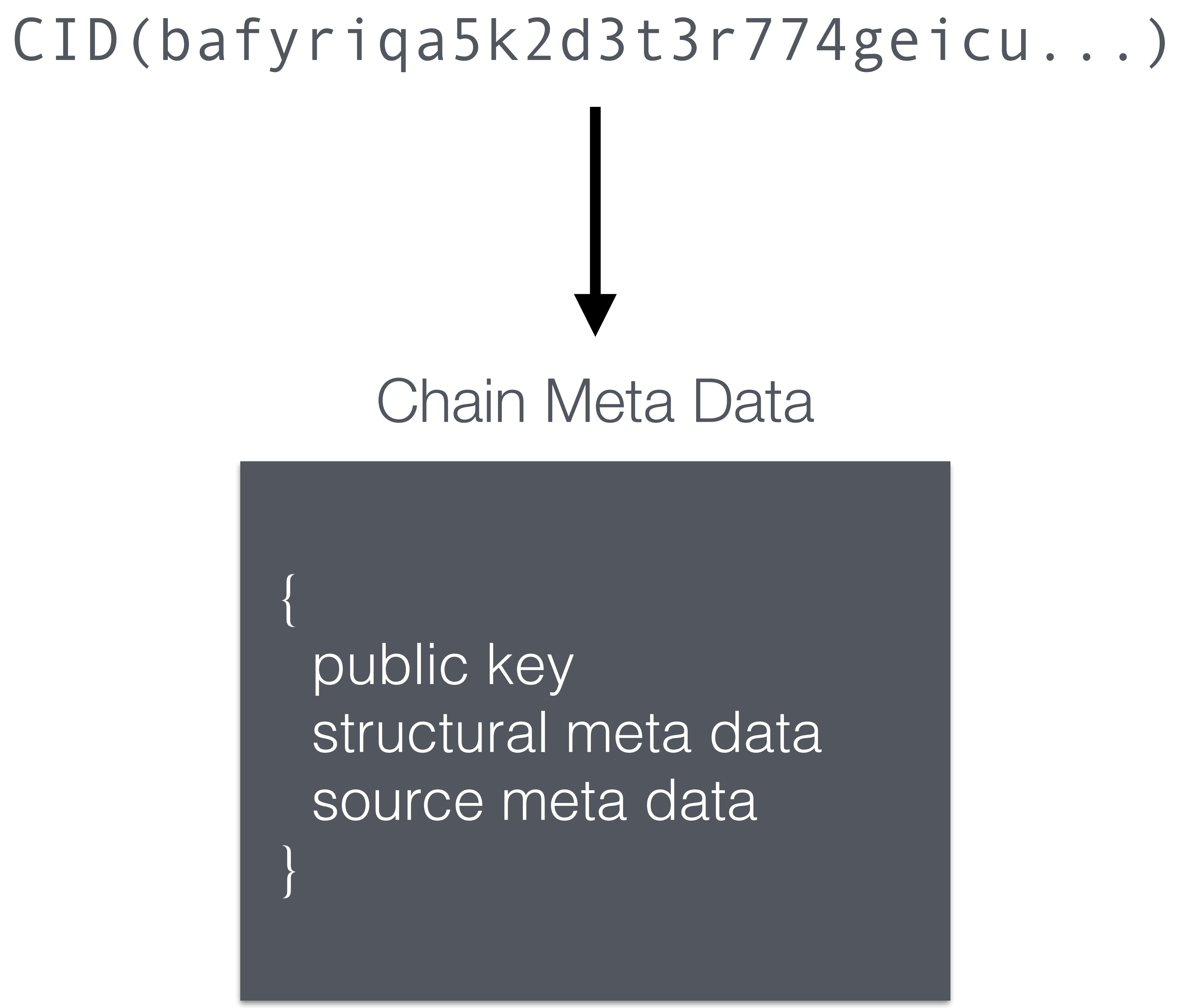
```
0x12326A858049CB305292AA7084FAB18A5DC6
      6FC8B3594167FAE4CC43663B4AF36595
```

HASH DIGEST

```
0x12 = sha2-256
0x32 = 256 bits
```

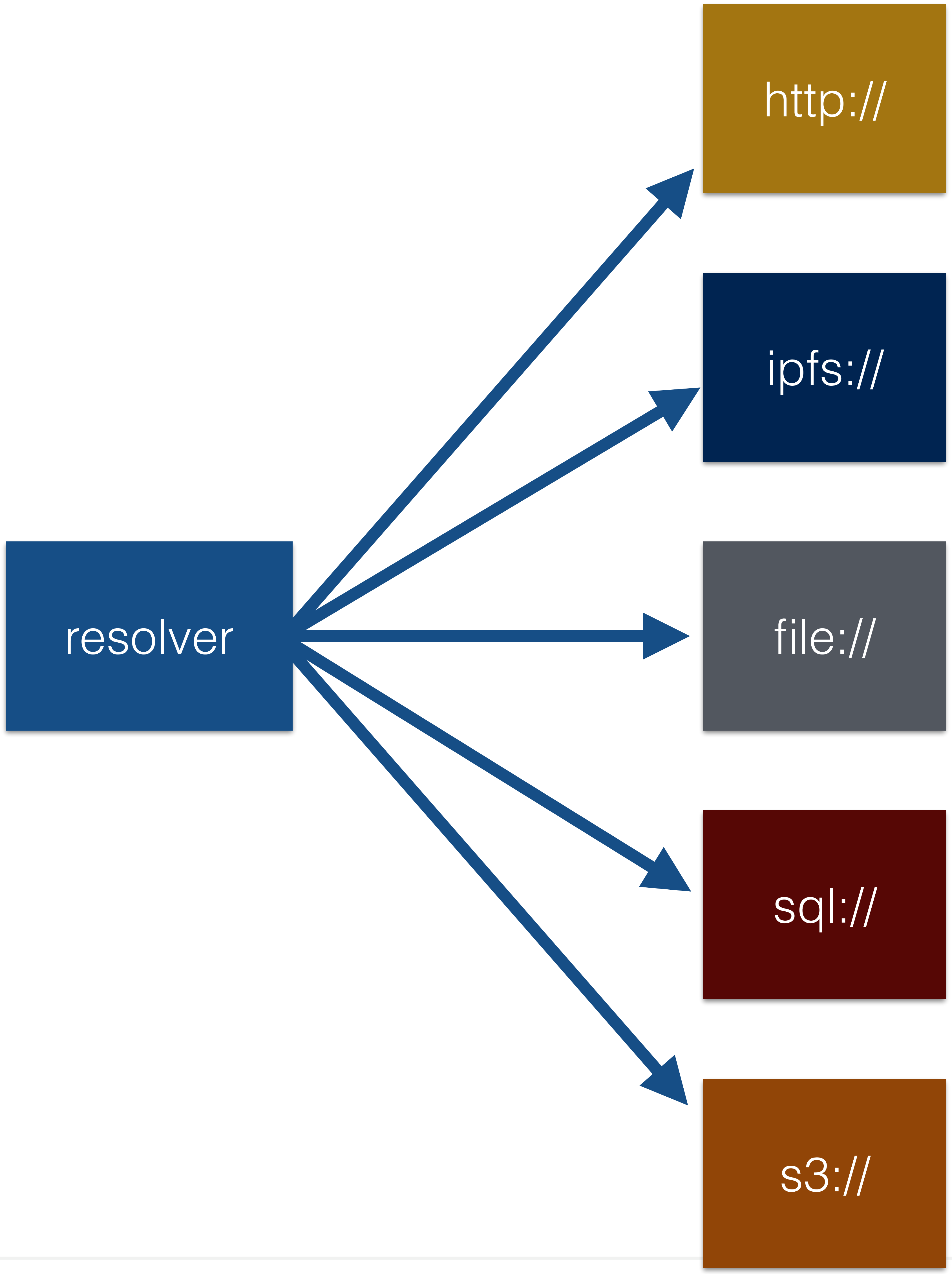
Features that make Twine flexible

- Generic payloads and sub-protocols.
- Content addressing and CIDs (hashes are the identifier)
- Chain data structures associate chains with CIDs



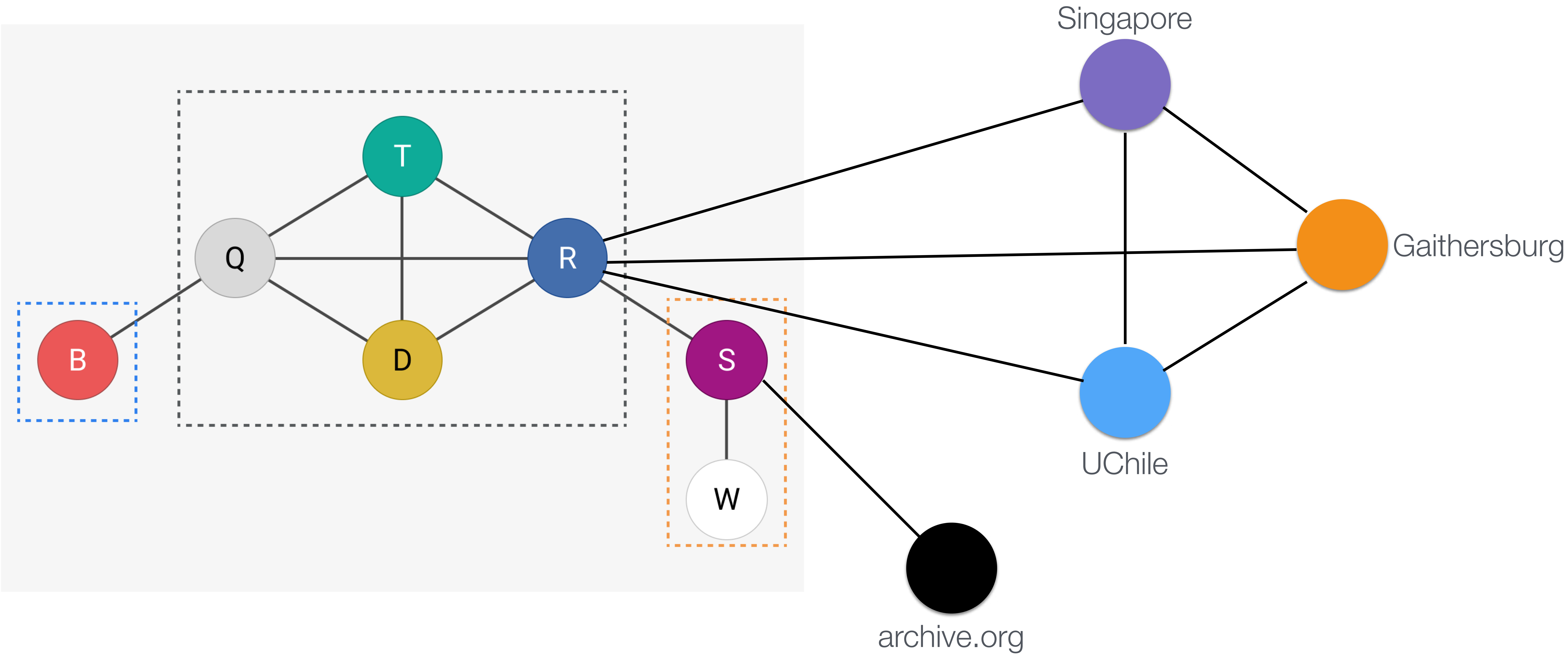
Features that make Twine flexible

- Generic payloads and sub-protocols.
- Content addressing and CIDs (hashes are the identifier)
- Chain data structures associate chains with CIDs
- Flexible storage options



Possible futures

"Cat's Cradle" is more fun with friends



Thanks

Here are some key takeaways

- Twine is still in development but very fleshed out and easy to develop with
- CURBy is the first example use-case but hopefully the first of many
- Welcome interest for testing it out and or solving outstanding questions
  jasper.palfree@colorado.edu