A Halloween-themed night scene with a dark purple background. At the top center is a large, bright yellow full moon. Scattered across the sky are several white, five-pointed stars of varying sizes. On the left side, three black bat silhouettes are flying. On the right side, three more black bat silhouettes are flying. In the bottom left corner, there is a black silhouette of a graveyard with several tombstones of different shapes and sizes. In the bottom right corner, there is a black silhouette of a cat standing on a small hill, and next to it is a black silhouette of a jack-o'-lantern with a carved face. Two black, gnarled tree silhouettes with twisted branches are positioned on the left and right sides of the bottom. The text "Yikes! Moles!" is written in a white, cursive font in the center of the image.

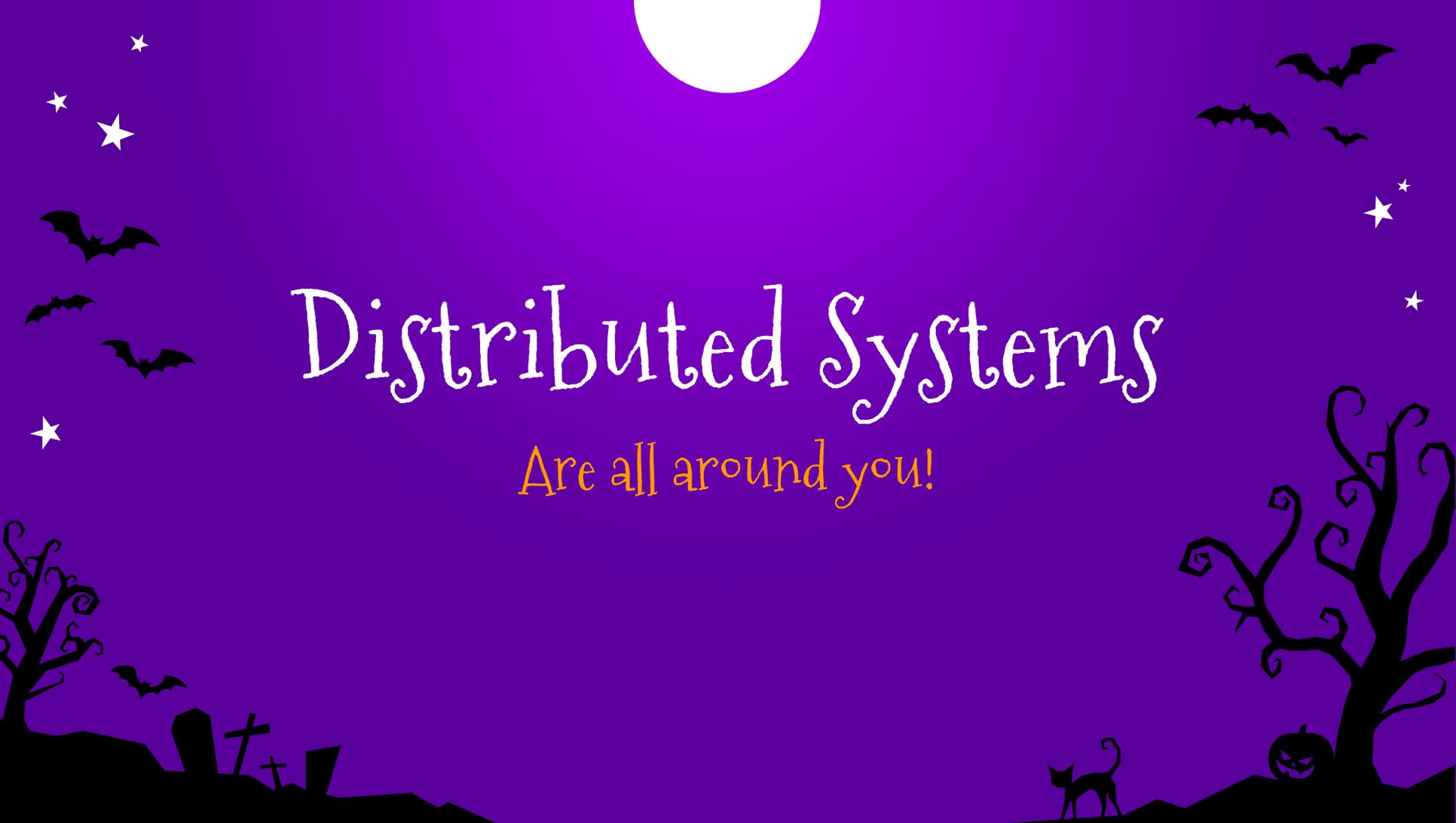
Yikes!
Moles!

CRDTs

Data types designed for use in replicated systems.

They provide eventual consistency.

There are two kinds.



Distributed Systems

Are all around you!



How Distributed Systems
Go BAD.



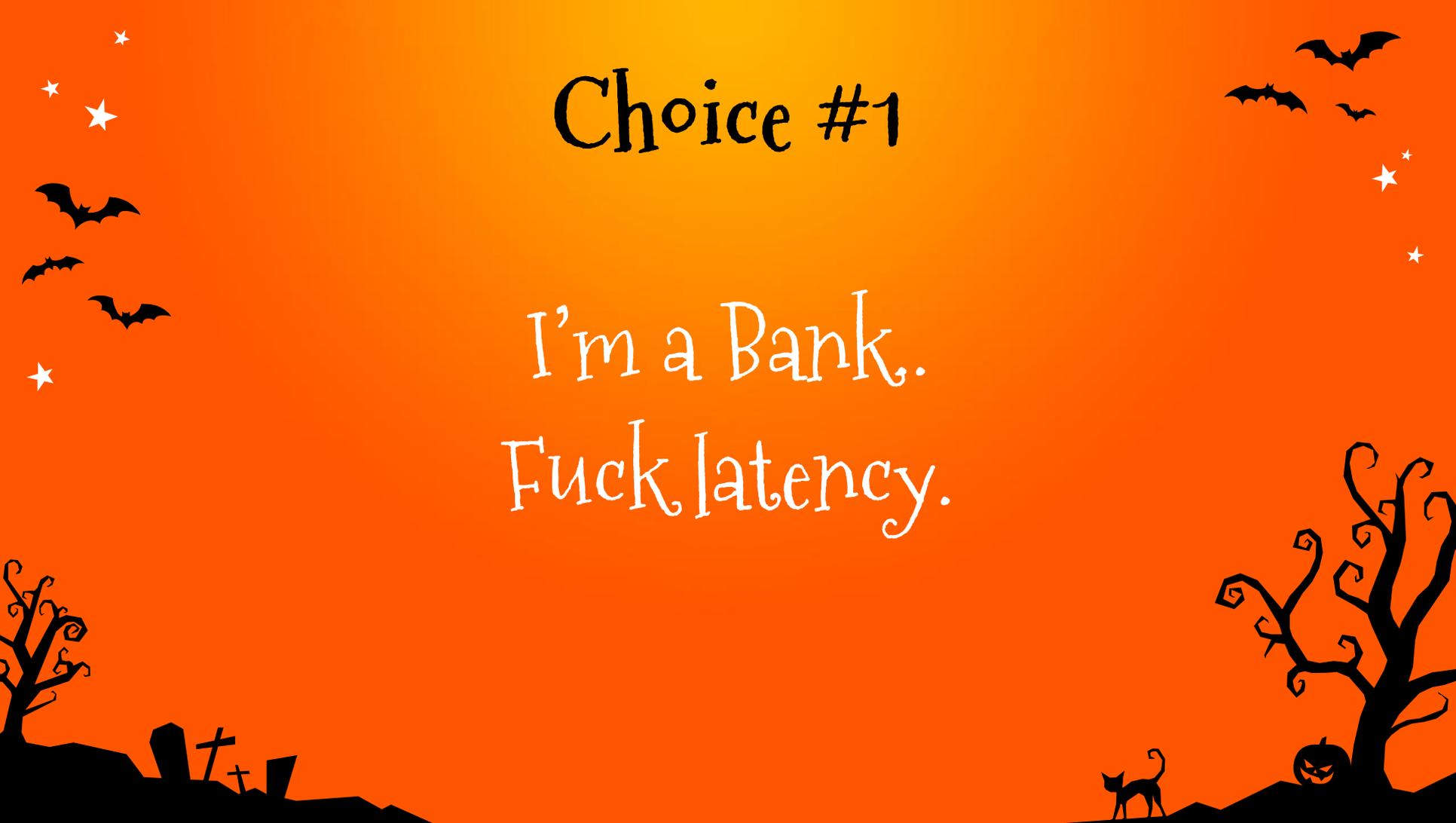
They forget
What you did
last summer.



(in) consistent.
(un) available – (late) nt.
Par-ti-tions!

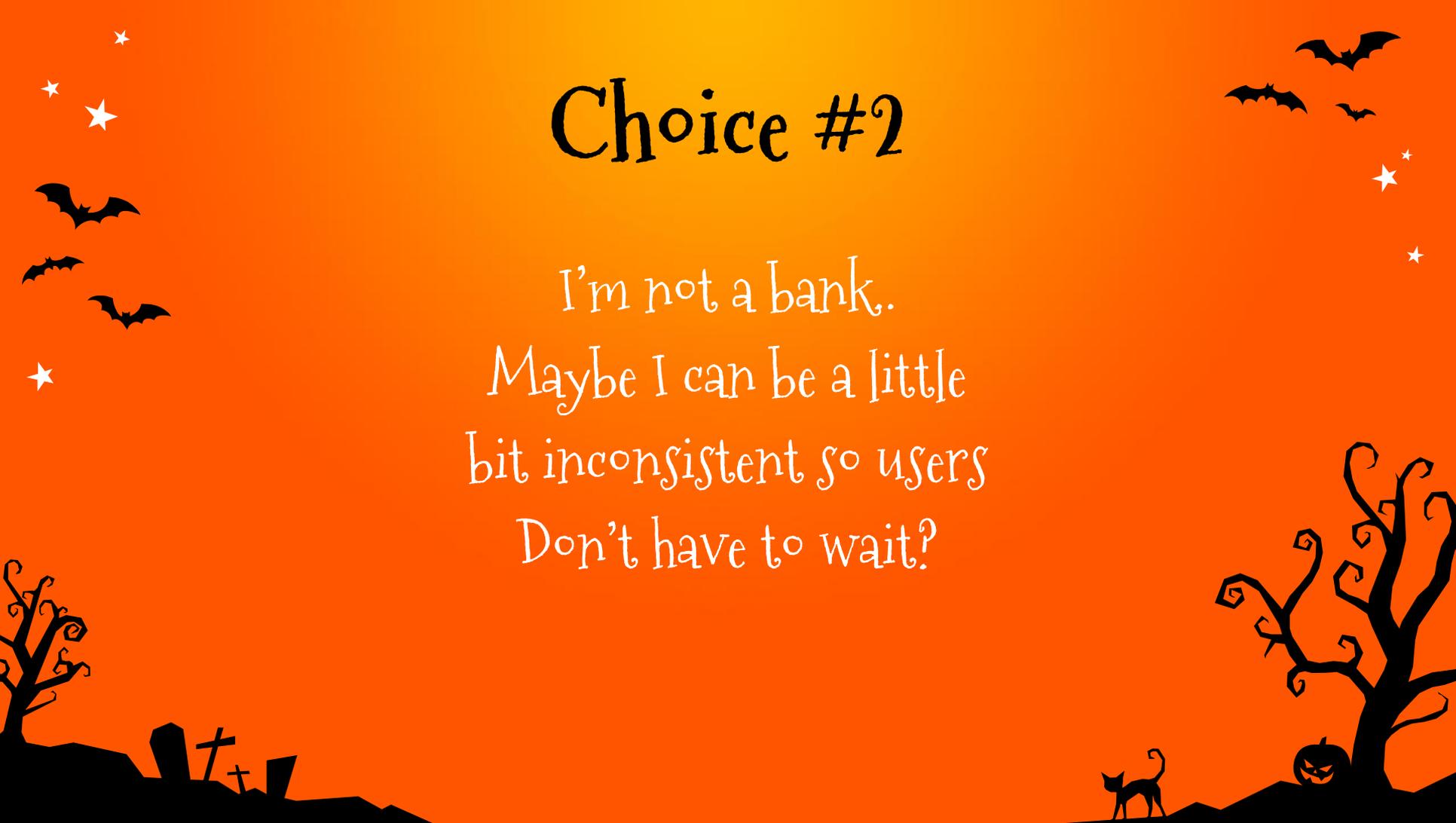
Choice #1

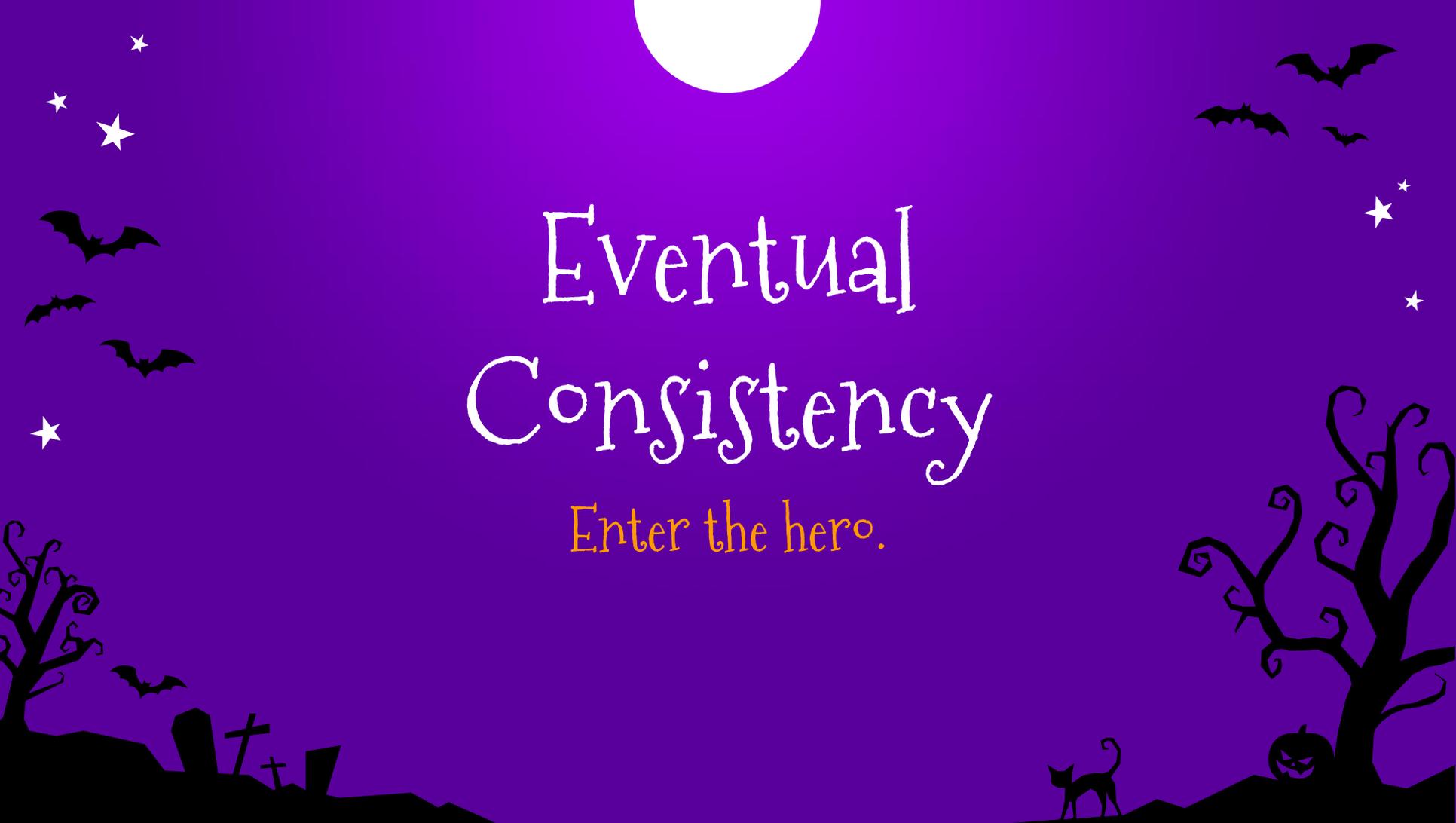
I'm a Bank.
Fuck latency.



Choice #2

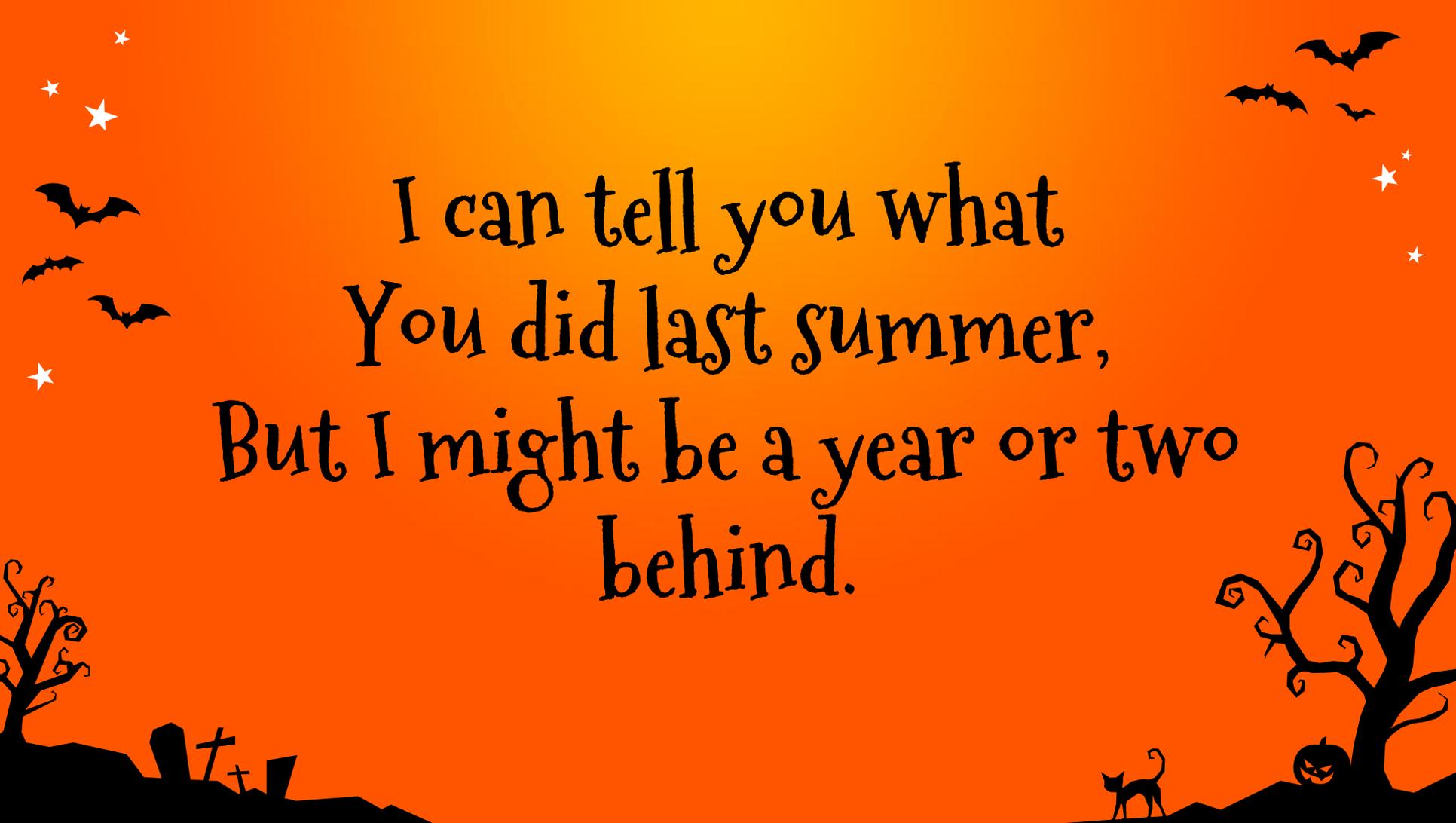
I'm not a bank.
Maybe I can be a little
bit inconsistent so users
Don't have to wait?





Eventual Consistency

Enter the hero.



I can tell you what
You did last summer,
But I might be a year or two
behind.

Conflicting Updates

I went to Hawai'i.

After that,
to Zanzibar.

After that,
to Thailand.

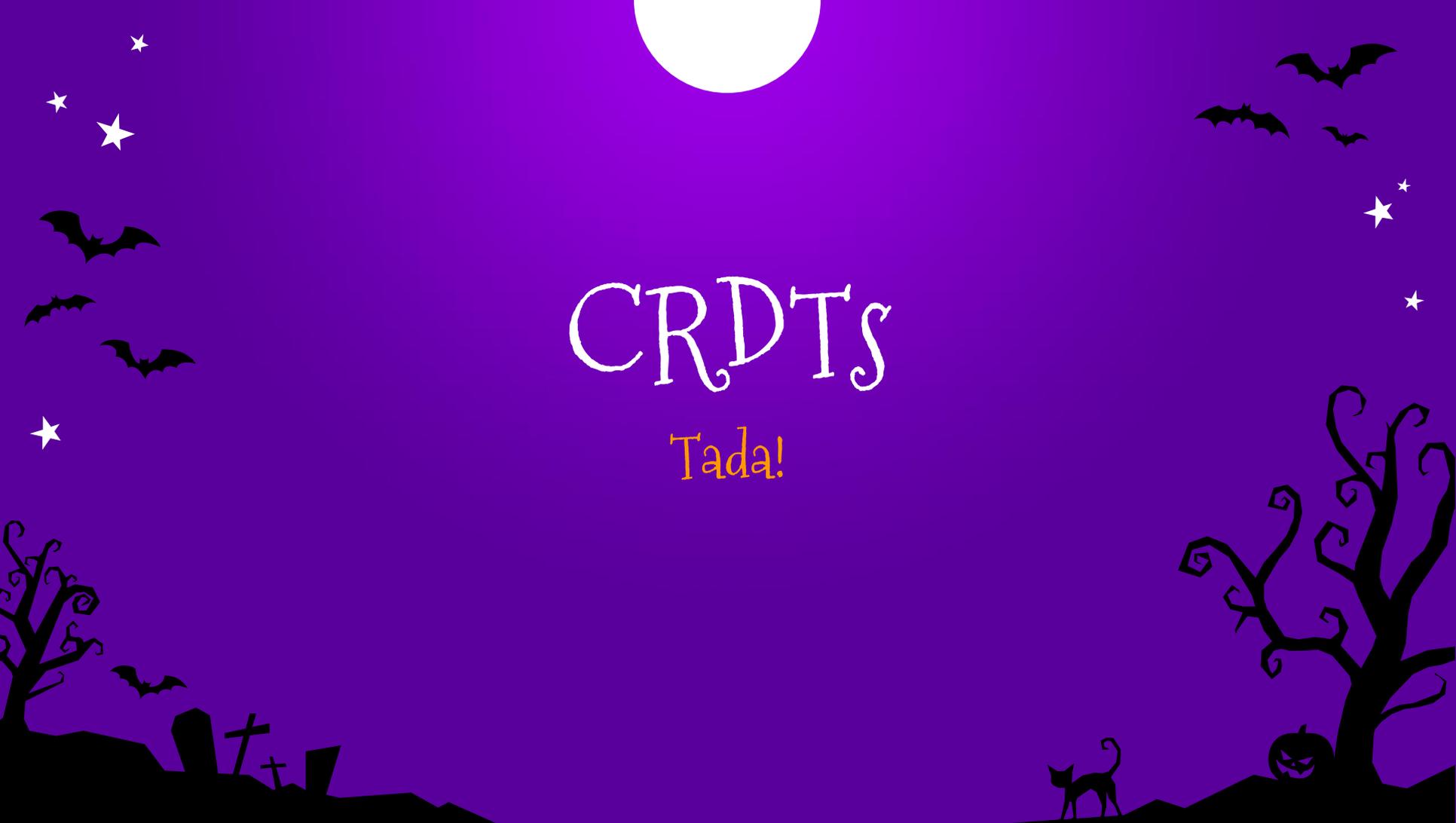


A little extra information...

I went to Hawai'i in 2014.

To Zanzibar in
2015.

To Thailand in
2016.



CRDTs

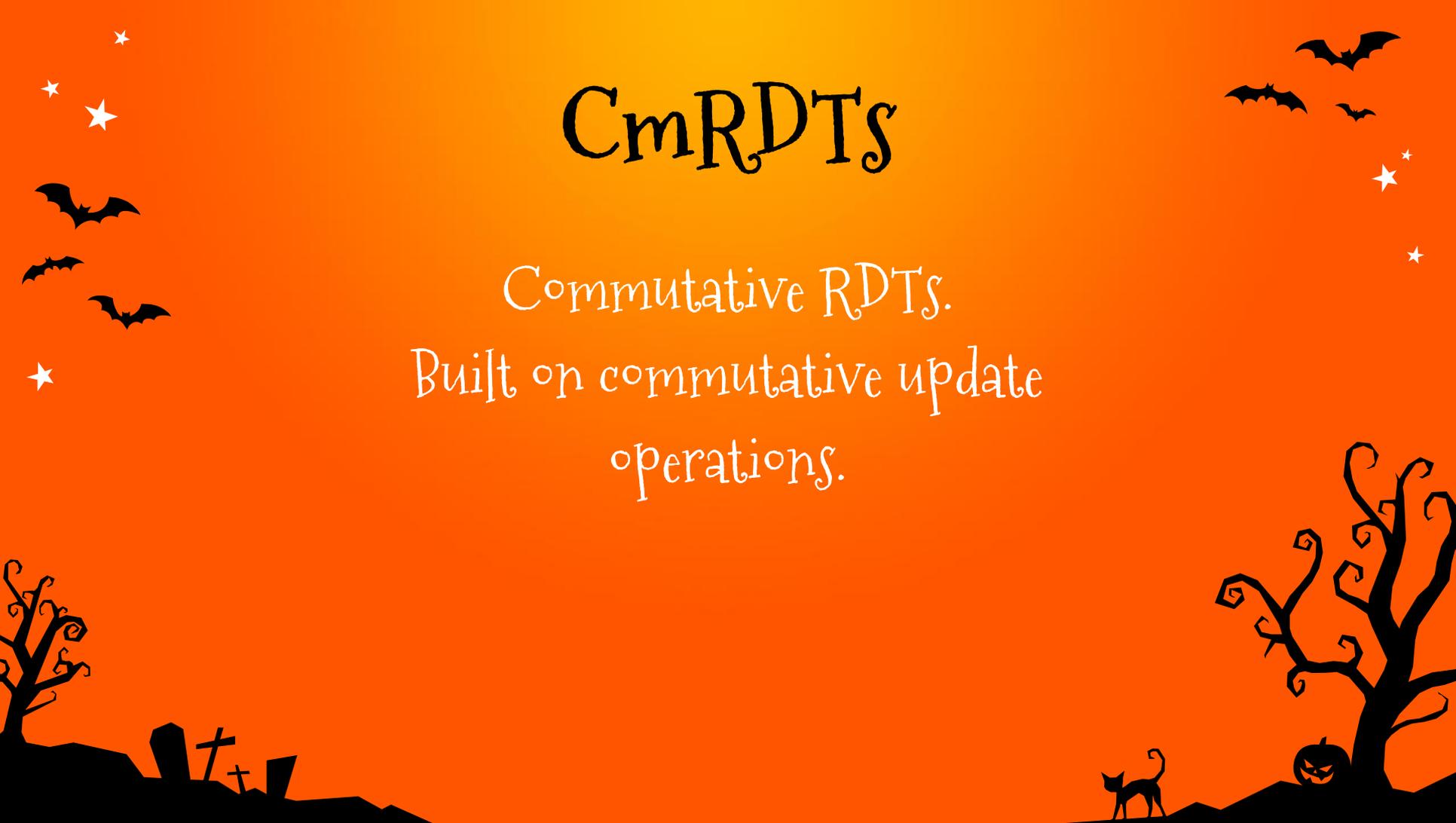
Tada!



Conflict-Free Replicated Data Type

CmRDTs

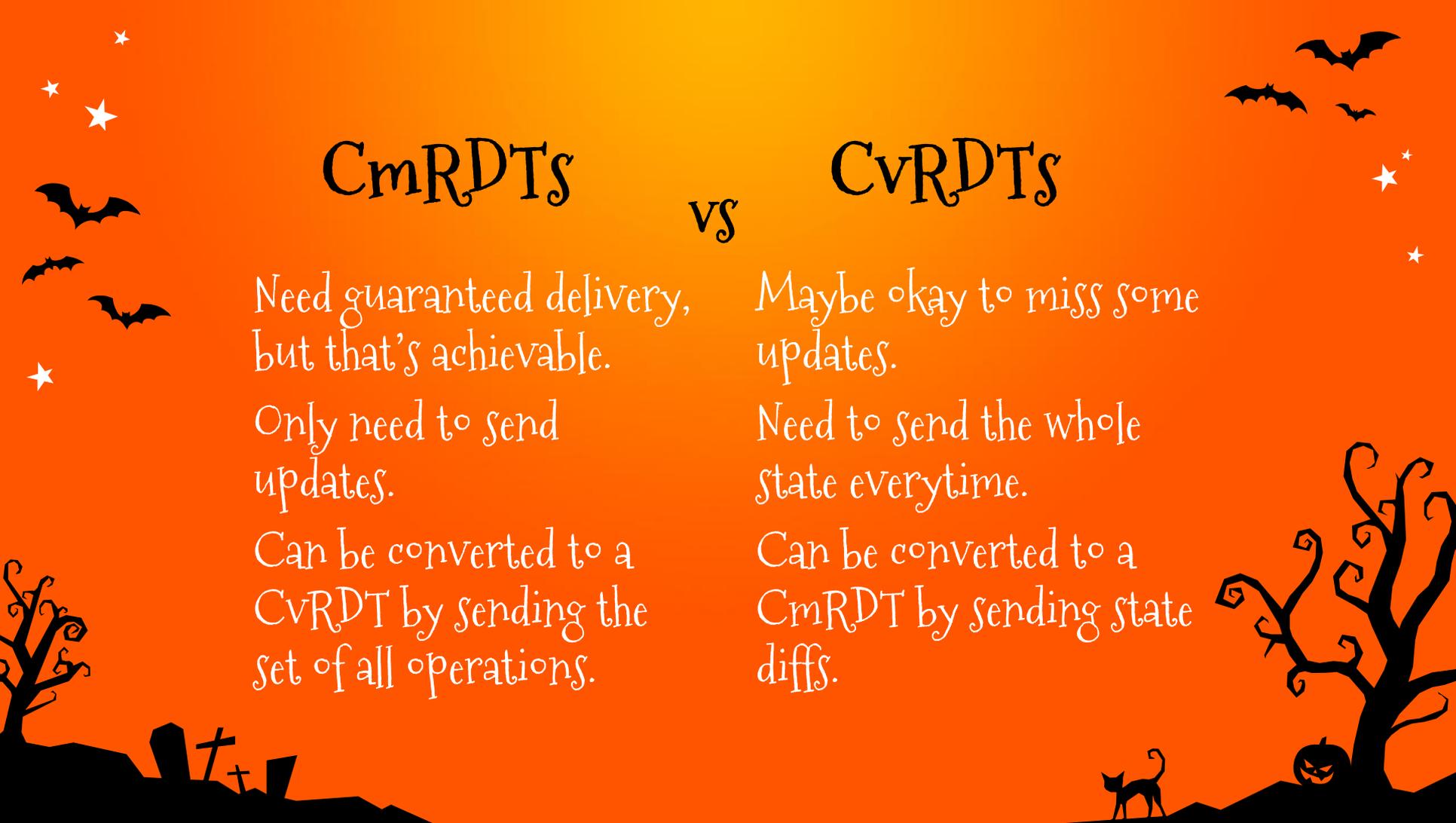
Commutative RDTs.
Built on commutative update
operations.



CvRDTs

Convergent RDTs.
Built on convergent state
merges.





CmRDTs

vs

CvRDTs

Need guaranteed delivery,
but that's achievable.

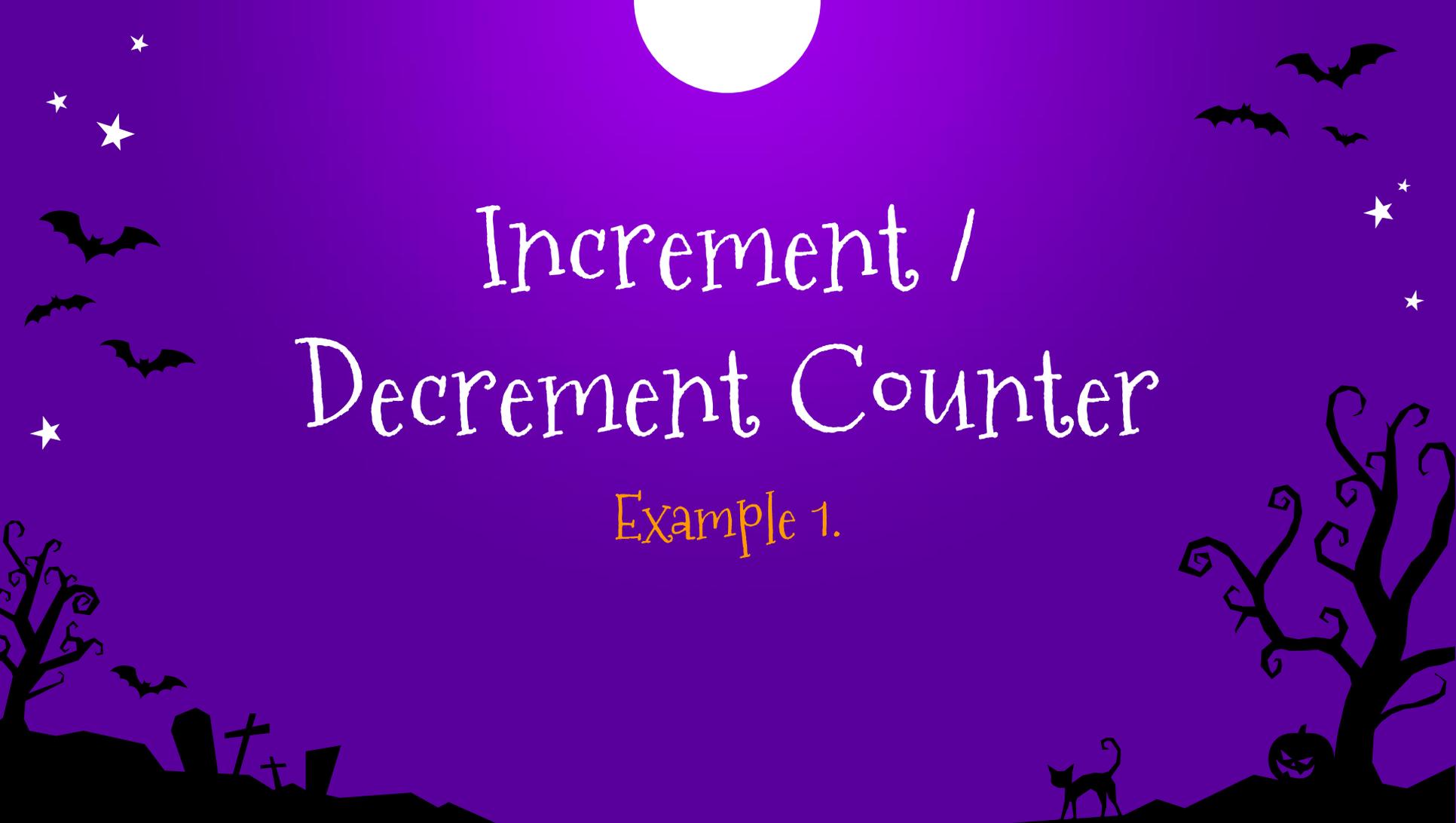
Only need to send
updates.

Can be converted to a
CvRDT by sending the
set of all operations.

Maybe okay to miss some
updates.

Need to send the whole
state everytime.

Can be converted to a
CmRDT by sending state
diffs.



Increment / Decrement Counter

Example 1.

CmRDT

Operation: $+X$.

Apply: Current $+X$.



CvRDT

State:

$$P = [o, o, o, \dots], N = [o, o, o, \dots]$$

Merge:

$$P_i = \max(P_i), N_i = \max(N_i)$$

Value:

$$V = \text{sum}(P_i) - \text{sum}(N_i)$$



2P Set

Example 2.

CmRDT

Operation: Add or remove x .

Apply:

Add X to A

or add X to R .

CvRDT

State:

$A: \emptyset, R: \emptyset$

Merge:

$A = A_1 \cup A_2, R = R_1 \cup R_2$

Value:

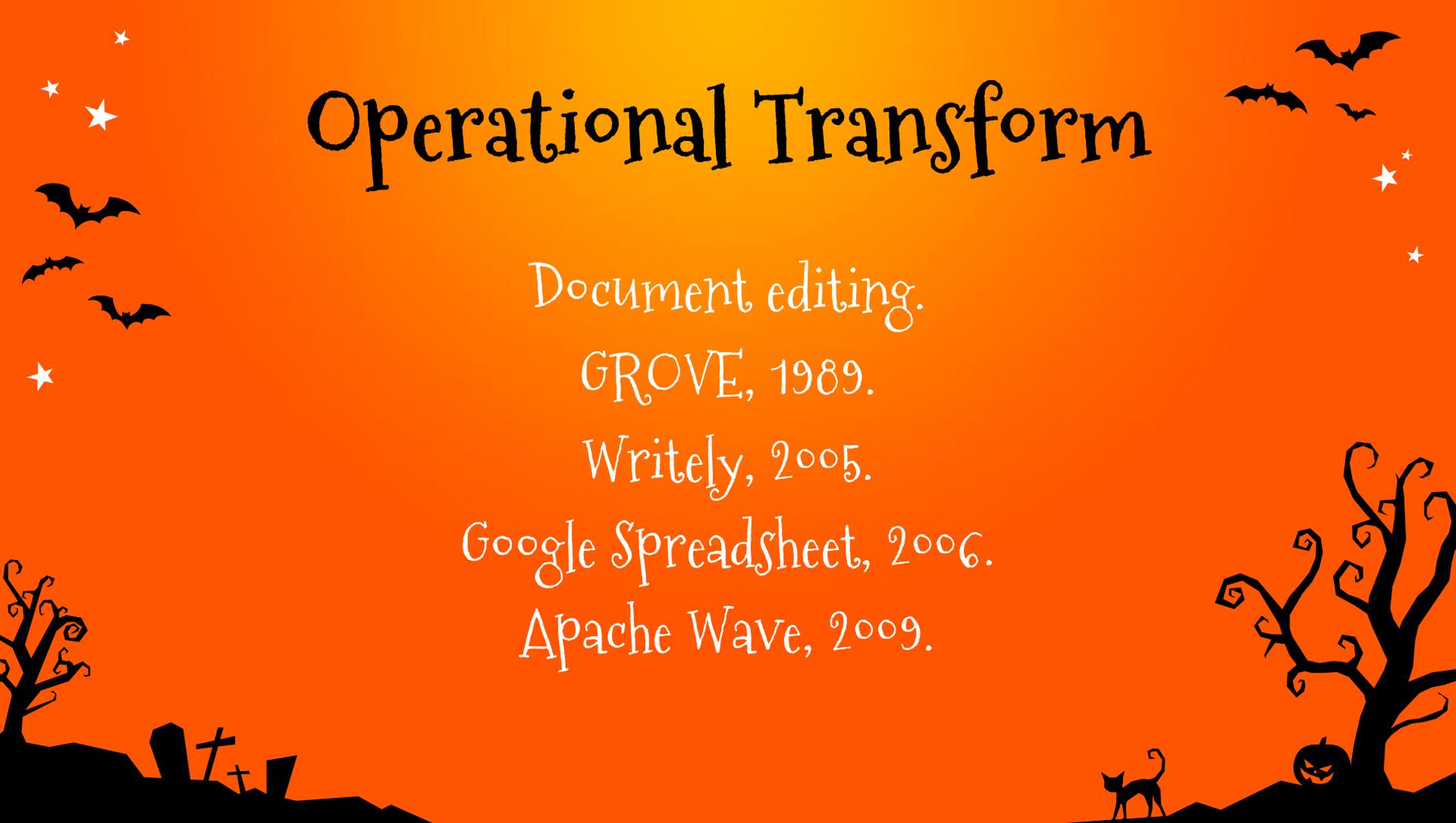
E in A but not in R



Ancient Lore

We are not the first
To dream such things.

Operational Transform



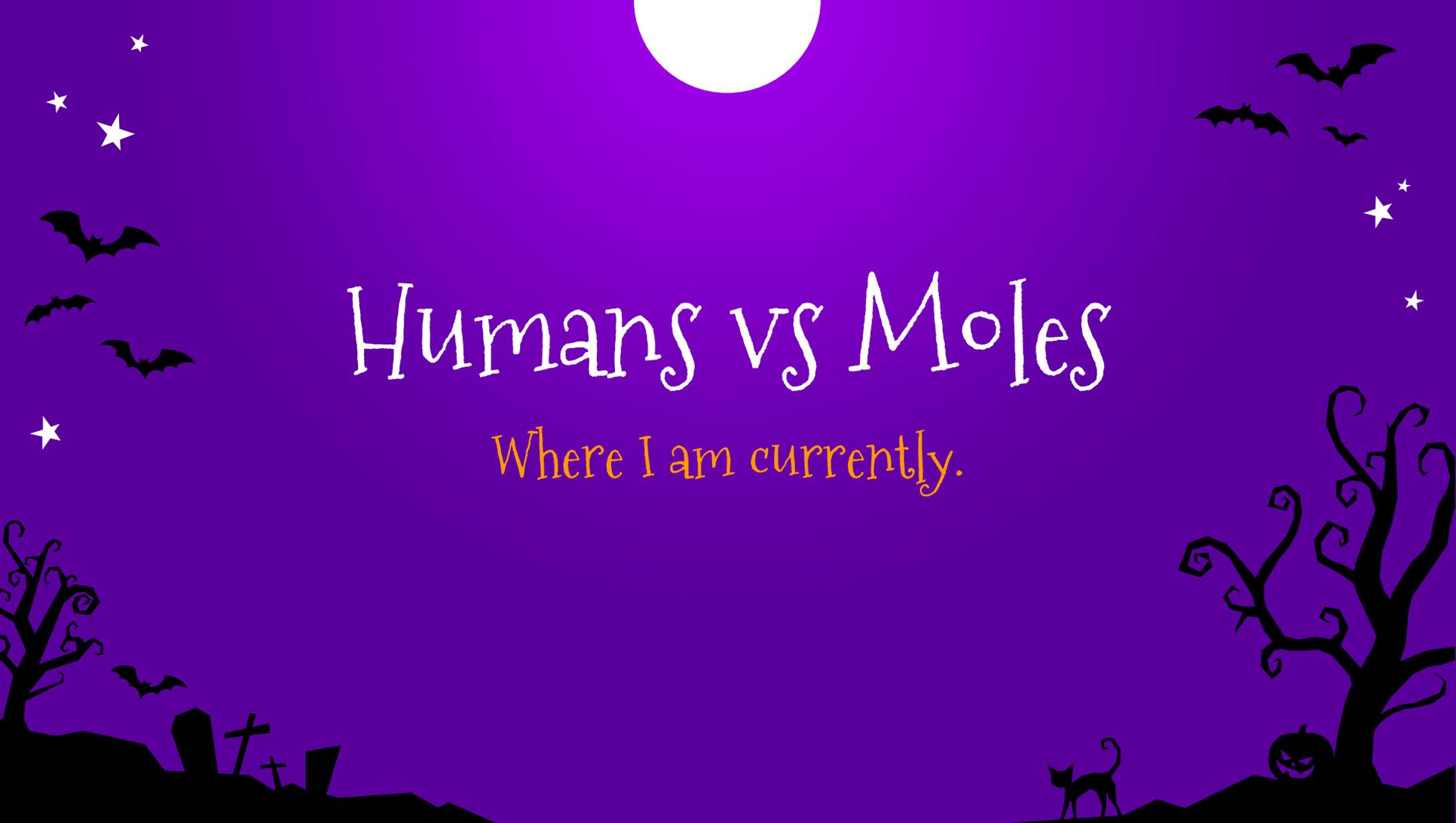
Document editing.

GROVE, 1989.

Writely, 2005.

Google Spreadsheet, 2006.

Apache Wave, 2009.

A Halloween-themed background with a dark purple sky. At the top center is a large, bright white full moon. Scattered throughout the sky are several white stars of varying sizes. Black silhouettes of bats are flying in various directions. In the foreground, there are black silhouettes of a graveyard with several tombstones of different shapes and sizes. On the right side, there is a large, gnarled tree with twisted branches. At the bottom right, there is a black silhouette of a cat and a jack-o'-lantern with a carved face.

Humans vs Moles

Where I am currently.



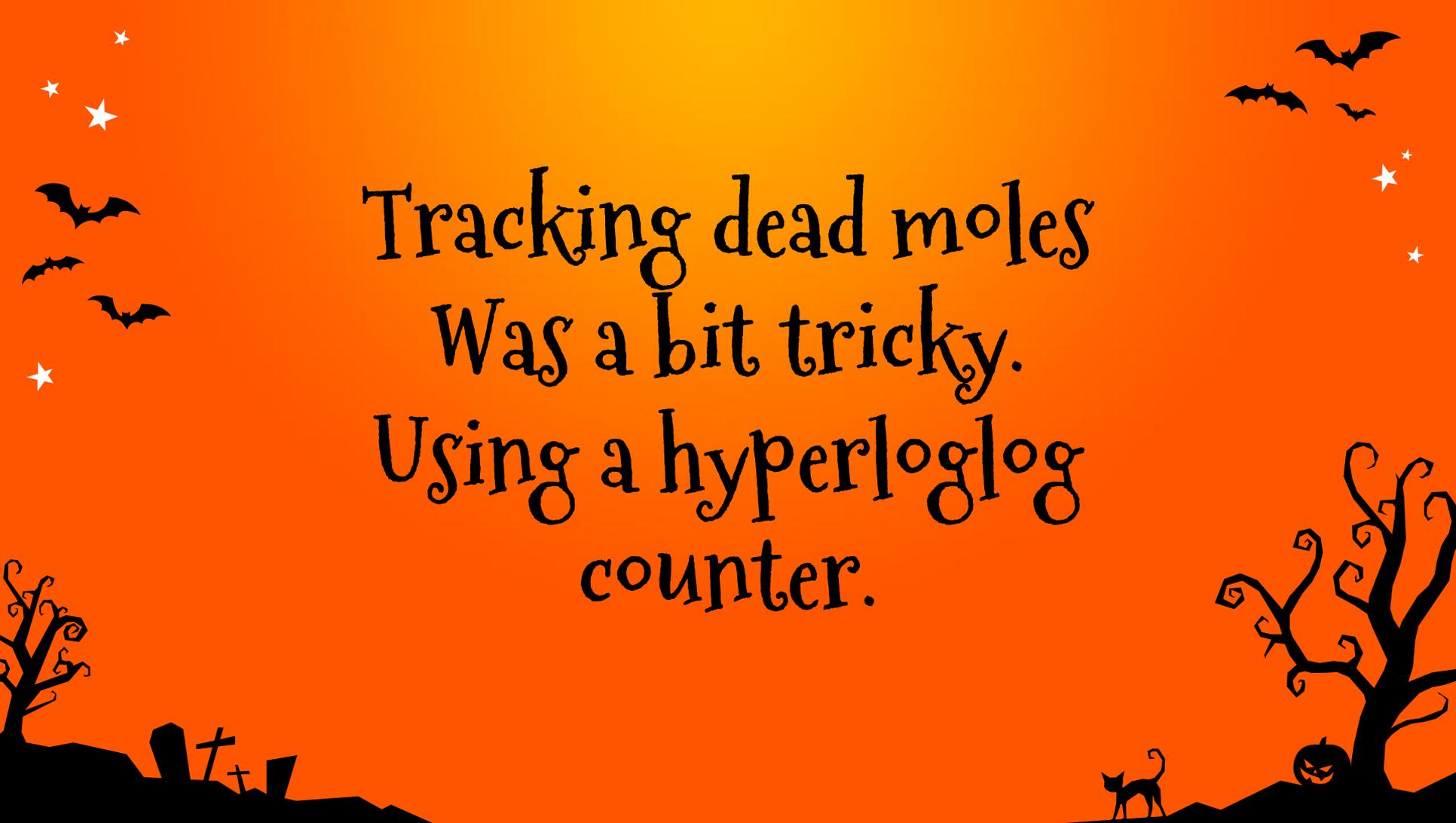
Pygame CFFI
Crossbar.io
Autobahn



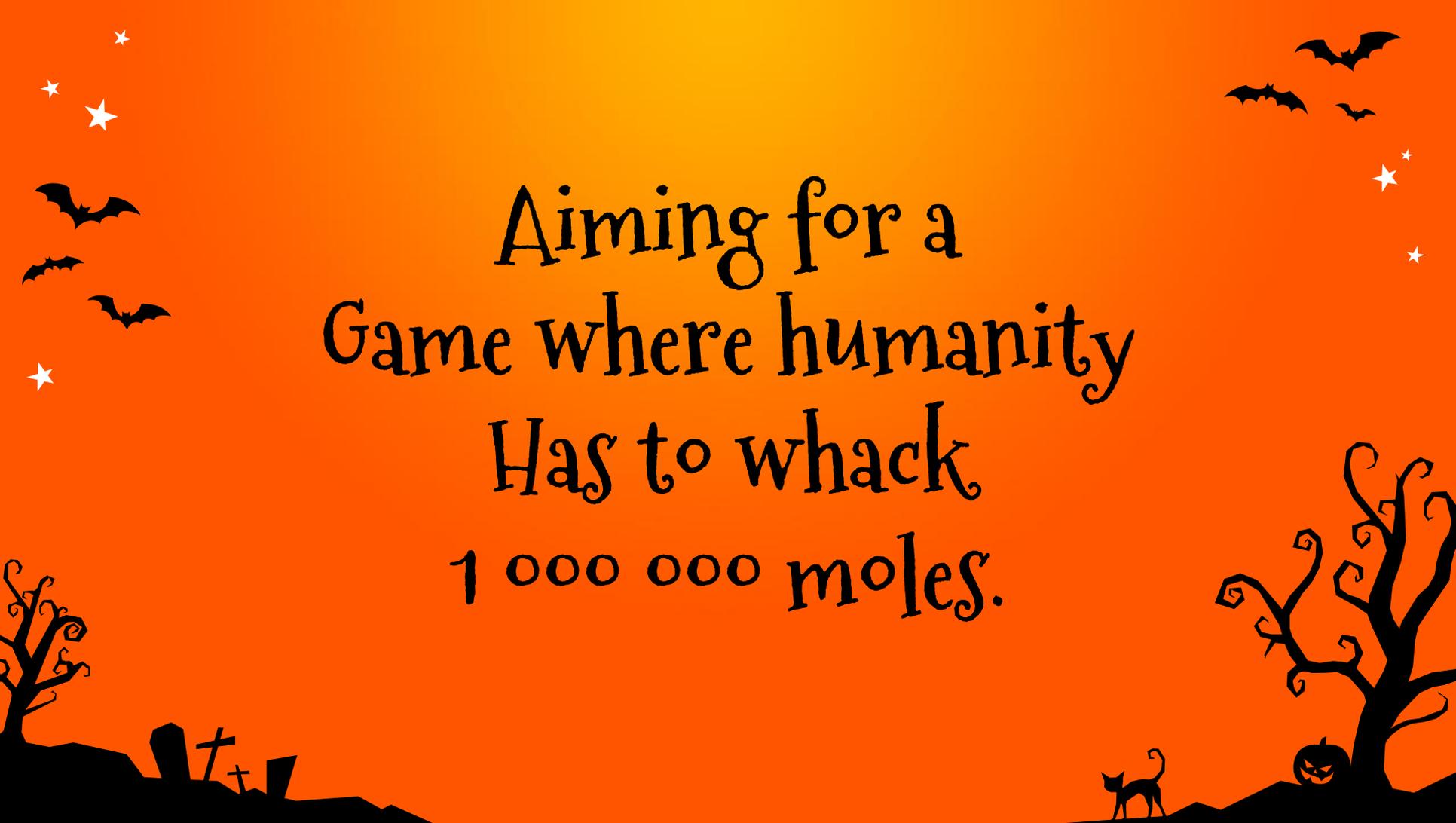
Using CvRDT so
New players don't
Need anything special.



Updates can be
Applied locally
Straight away.



Tracking dead moles
Was a bit tricky.
Using a hyperloglog
counter.



Aiming for a
Game where humanity
Has to whack
1 000 000 moles.