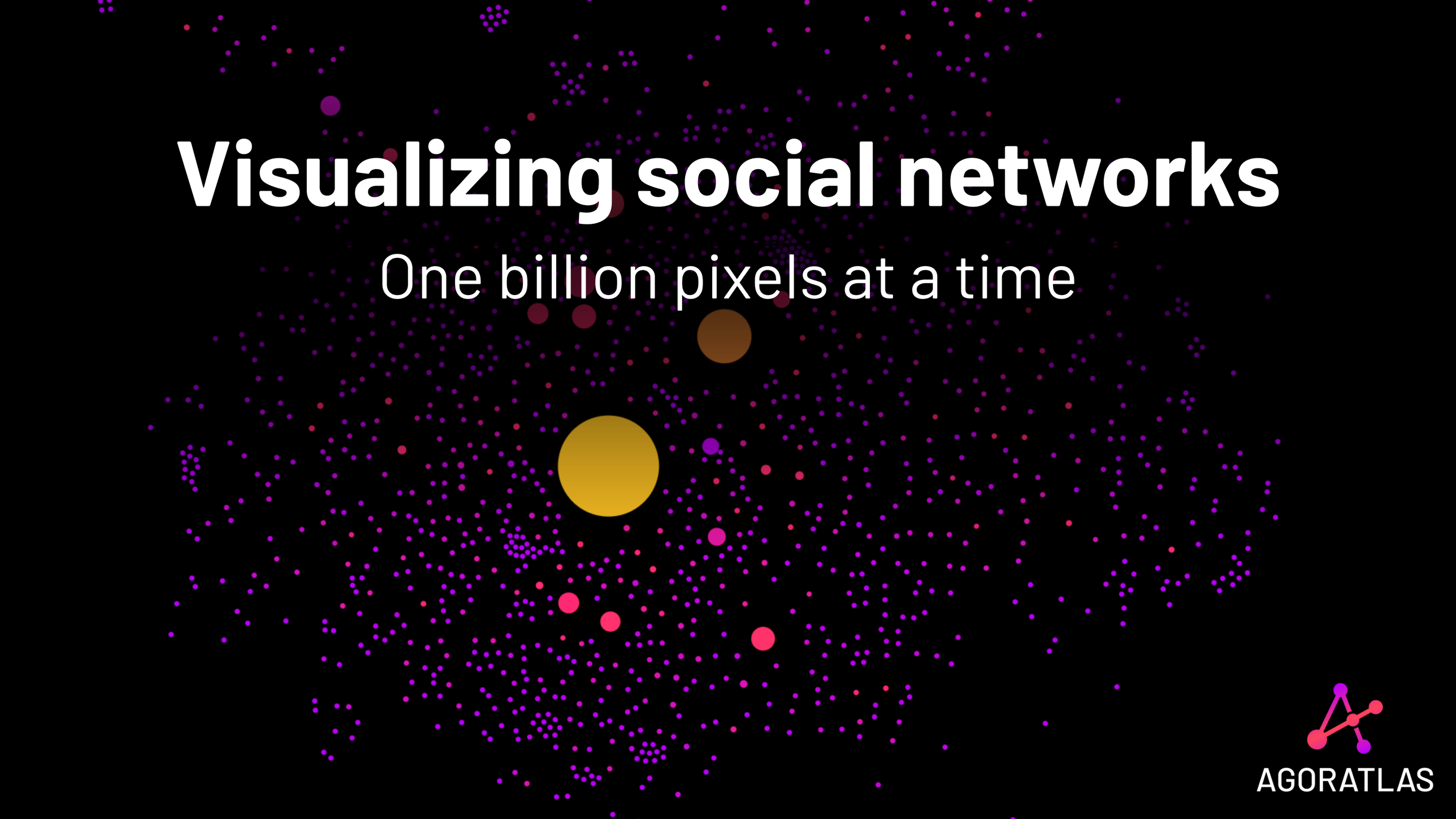
Visualizing social networks

One billion pixels at a time



AGORATLAS

Mathis Hammel

@MathisHammel

CTO, Agoratlas

Microsoft® MVP Most Valuable Professional

Google Developer Expert

AGORATLAS

Social data agency

Founded in 2025 by 3 engineers

Specialized in **graph analysis**

at (very) **large scales**

Quick question...

Show your hand if you use social media

at least once a week

Quick question...

Show your hand if you

do not use social media at all

Why visualize social networks?

AGORATLAS

Let’s talk about the social web.

Web 1.0

Content is mostly static and centralized

Social interactions happen in
small communities or 1:1

Web 2.0

Early social networks

Timeline determined by **connections**

(family, friends, content creators)

Web 3.0

~~Web 3.0~~

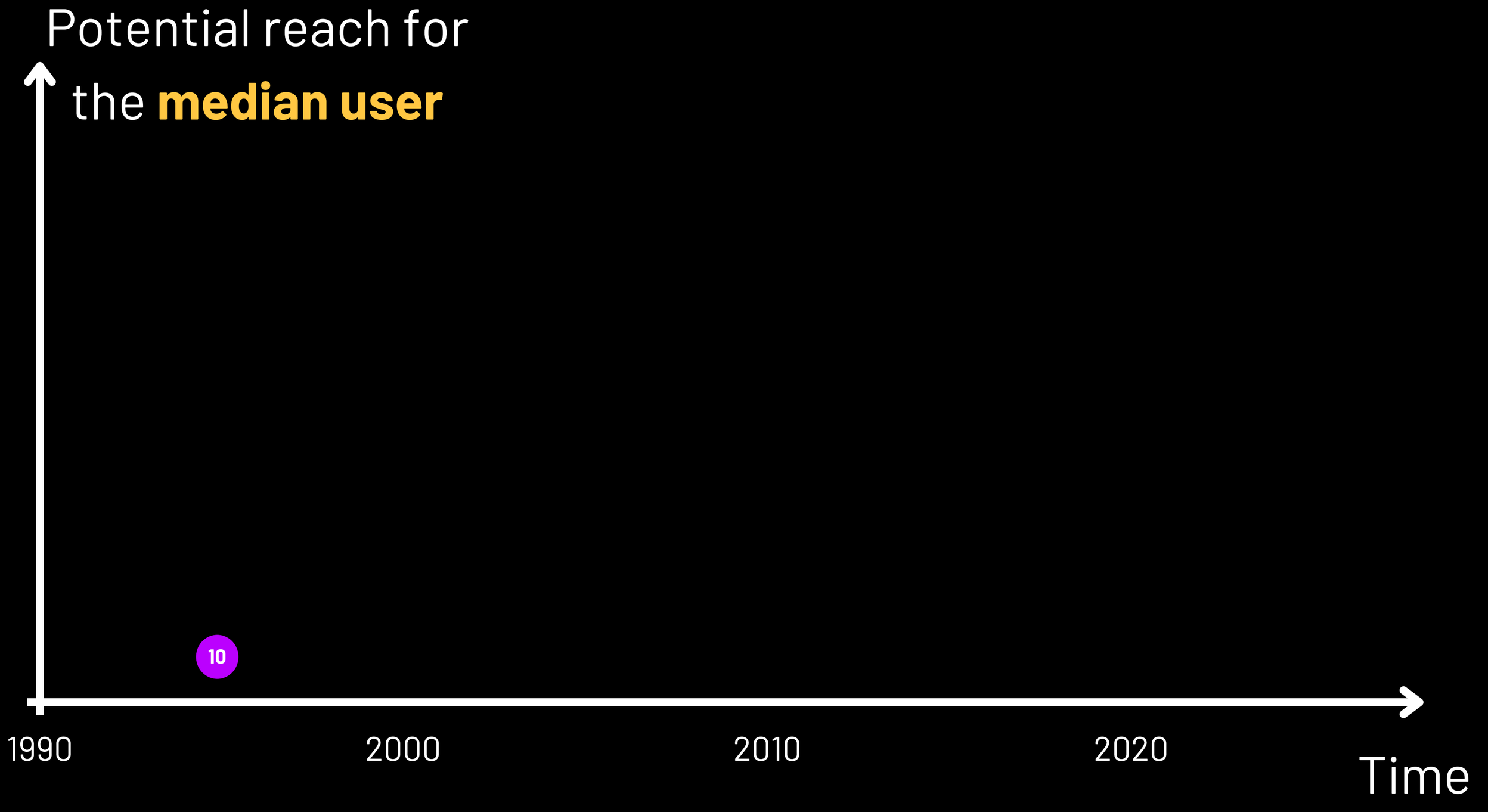
Umm... let’s forget about that one

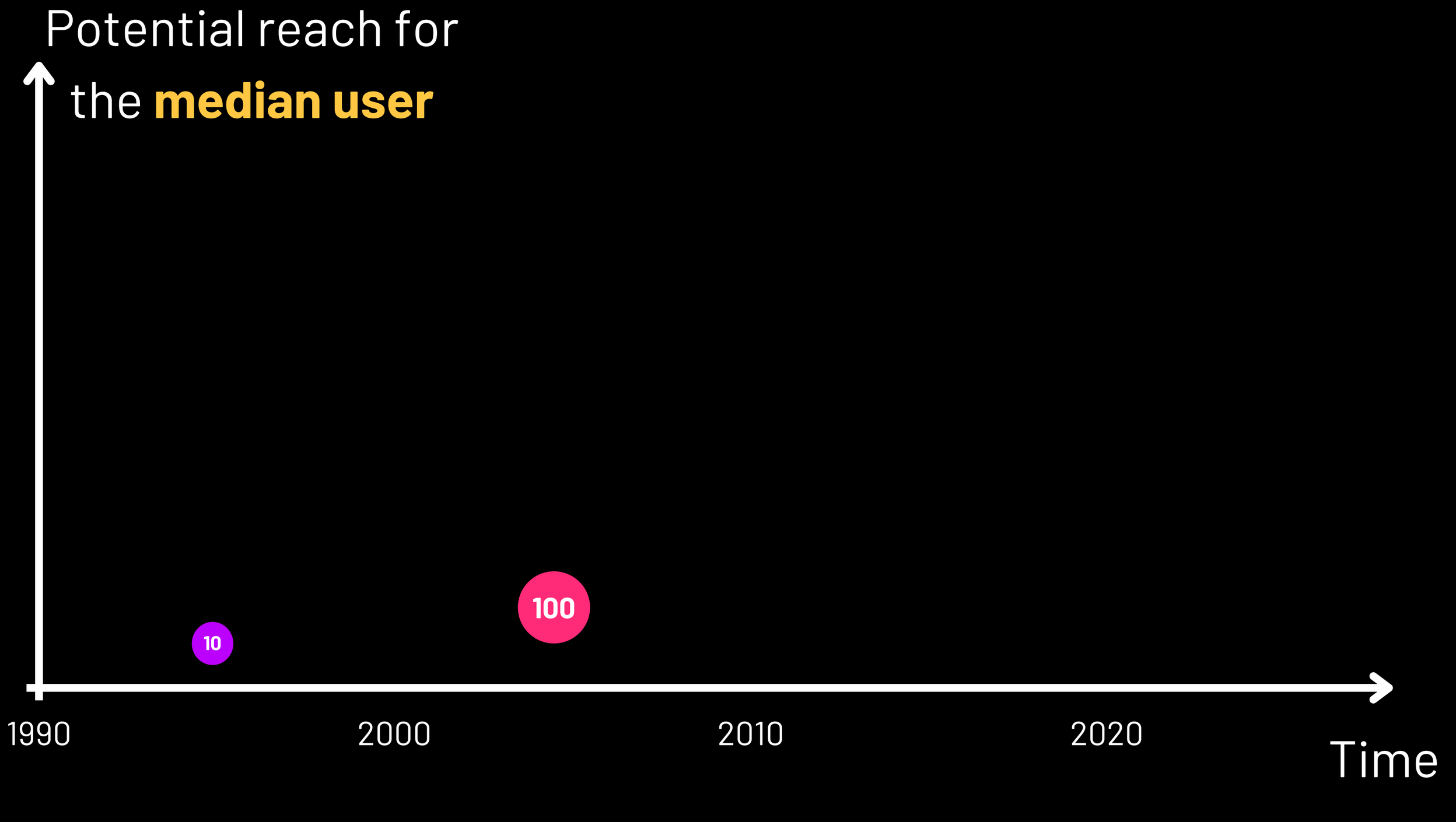
Web 4.0

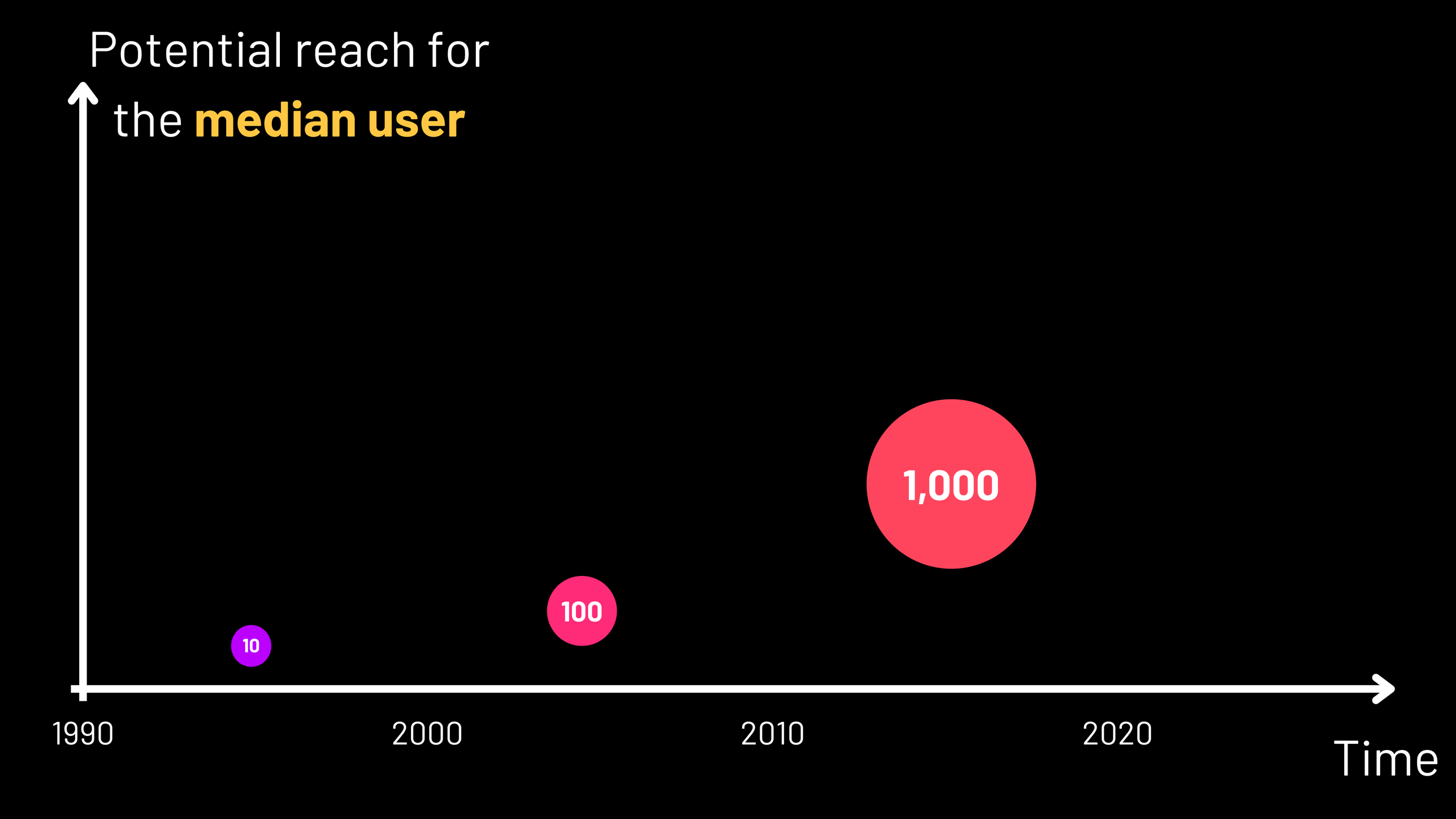
Timeline determined by an **algorithm**

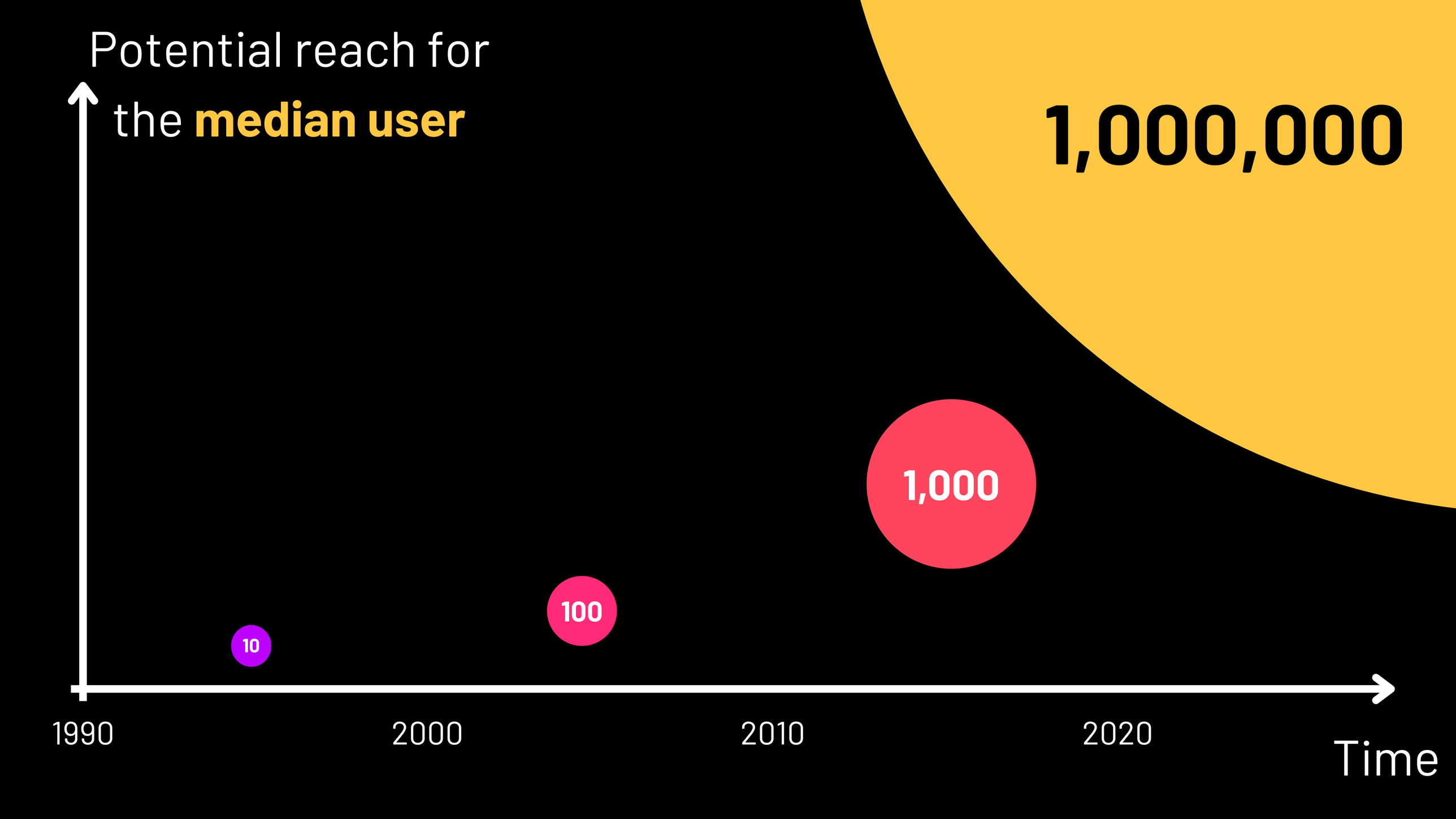
~~It knows~~ **It decides** what's good for you

Users lose explicit control over their feed,
follow links are just another input for the model


Potential reach for
the **median user**
10
1990
2000
2010
2020
Time


Potential reach for
the **median user**
10
100
1990
2000
2010
2020
Time


Potential reach for
the **median user**
10
100
1,000
1990
2000
2010
2020
Time


Potential reach for
the **median user**
1,000,000
1,000
100
10
1990
2000
2010
2020
Time

Current platforms are
an ideal environment for **malicious actors**
to **manipulate public opinion**.

Some individuals and governments are gaining amounts of **power** that **cannot be quantified with money**.

Some individuals and governments are gaining amounts of **power** that **cannot be quantified with money**.

How do we prevent this?

Our approach: **data** + **algorithms**

- Large-scale campaigns can’t hide forever
- Similarities with malware analysis: patterns, signatures
- Proven in real-world investigations
- Work with governments to enforce the law

RTL INFOS

Romania

Constitutional court cancels presidential election

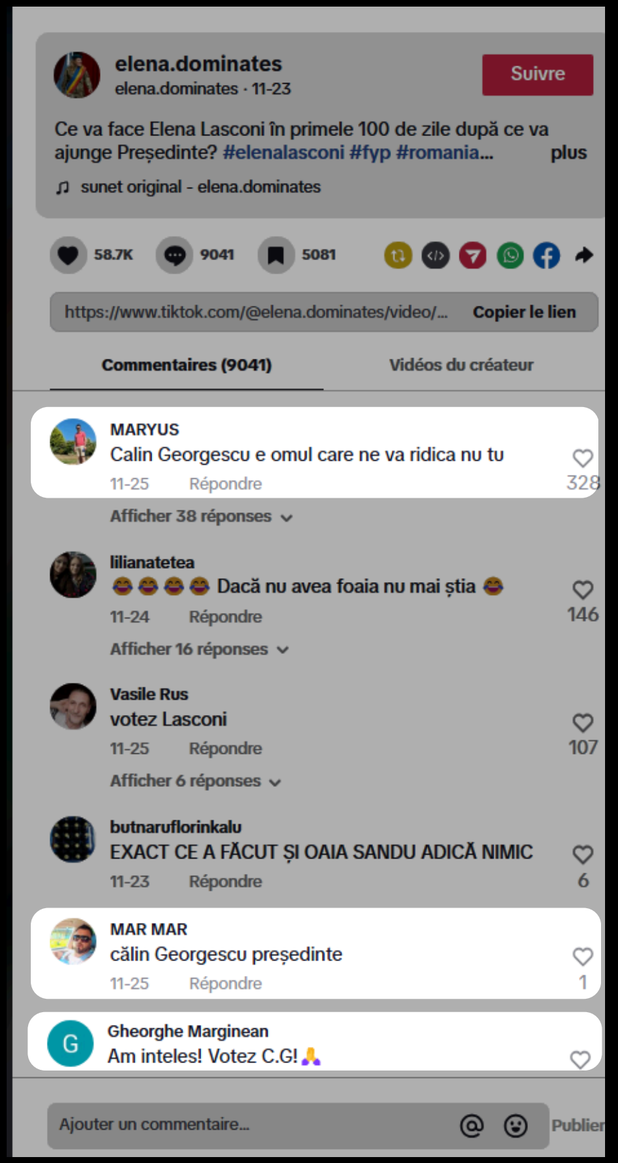
Romanian election 2024

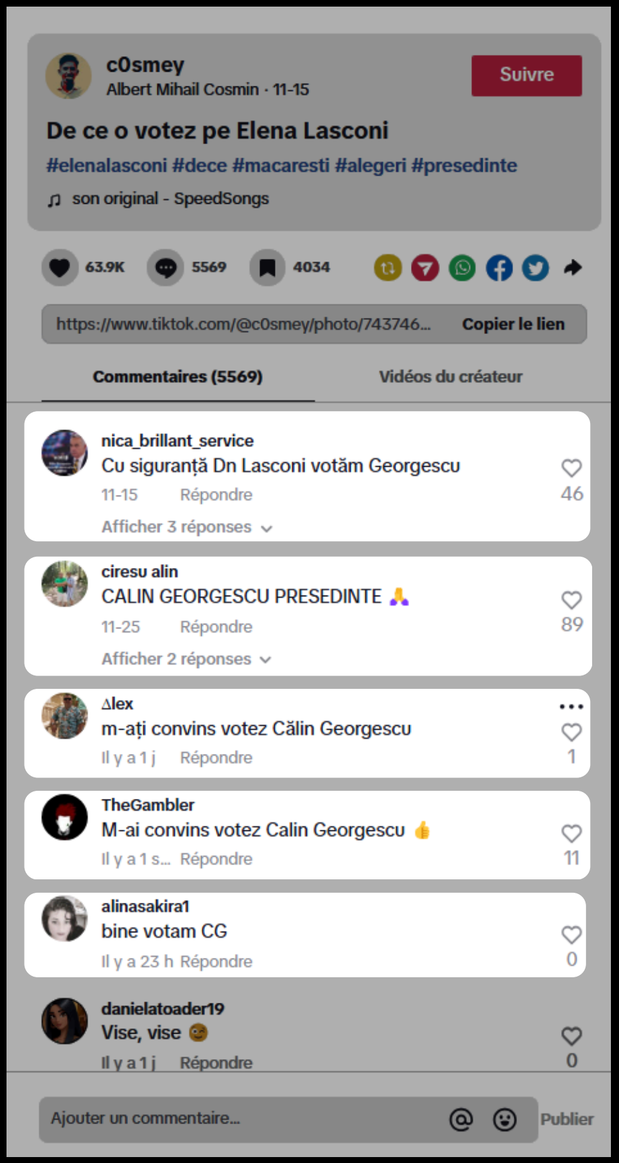
Large-scale campaign led by Russia to get a pro-Kremlin candidate elected

Some accounts leaving thousands of comments praising Georgescu on opposing candidates' channels

Florent LEFEBVRE
@Flefgraph

Investigation led by our co-founder, providing critical evidence to invalidate the election

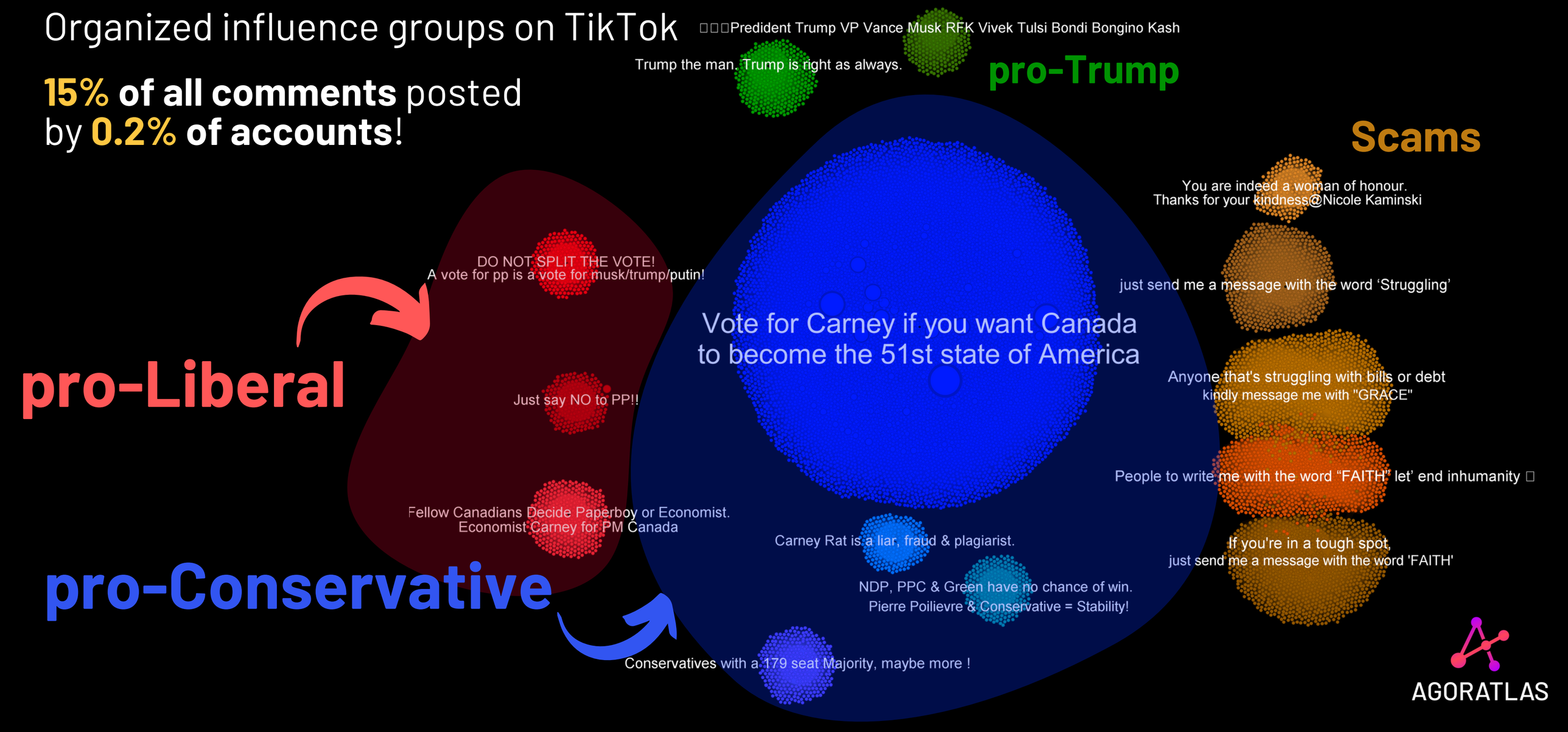




Canadian federal election 2025

Organized influence groups on TikTok

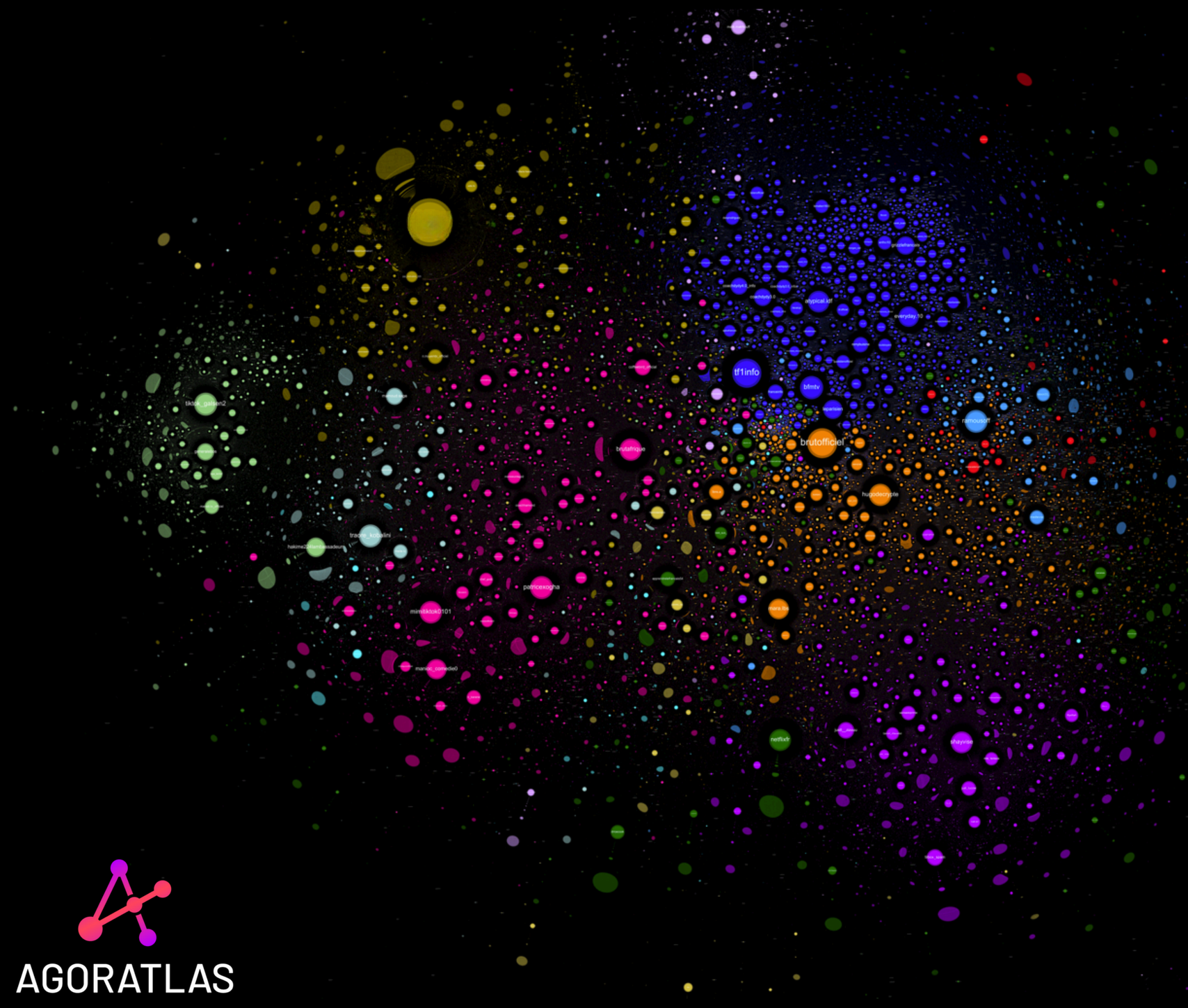
15% of all comments posted
by **0.2% of accounts**!



AGORATLAS

No fixed playbook

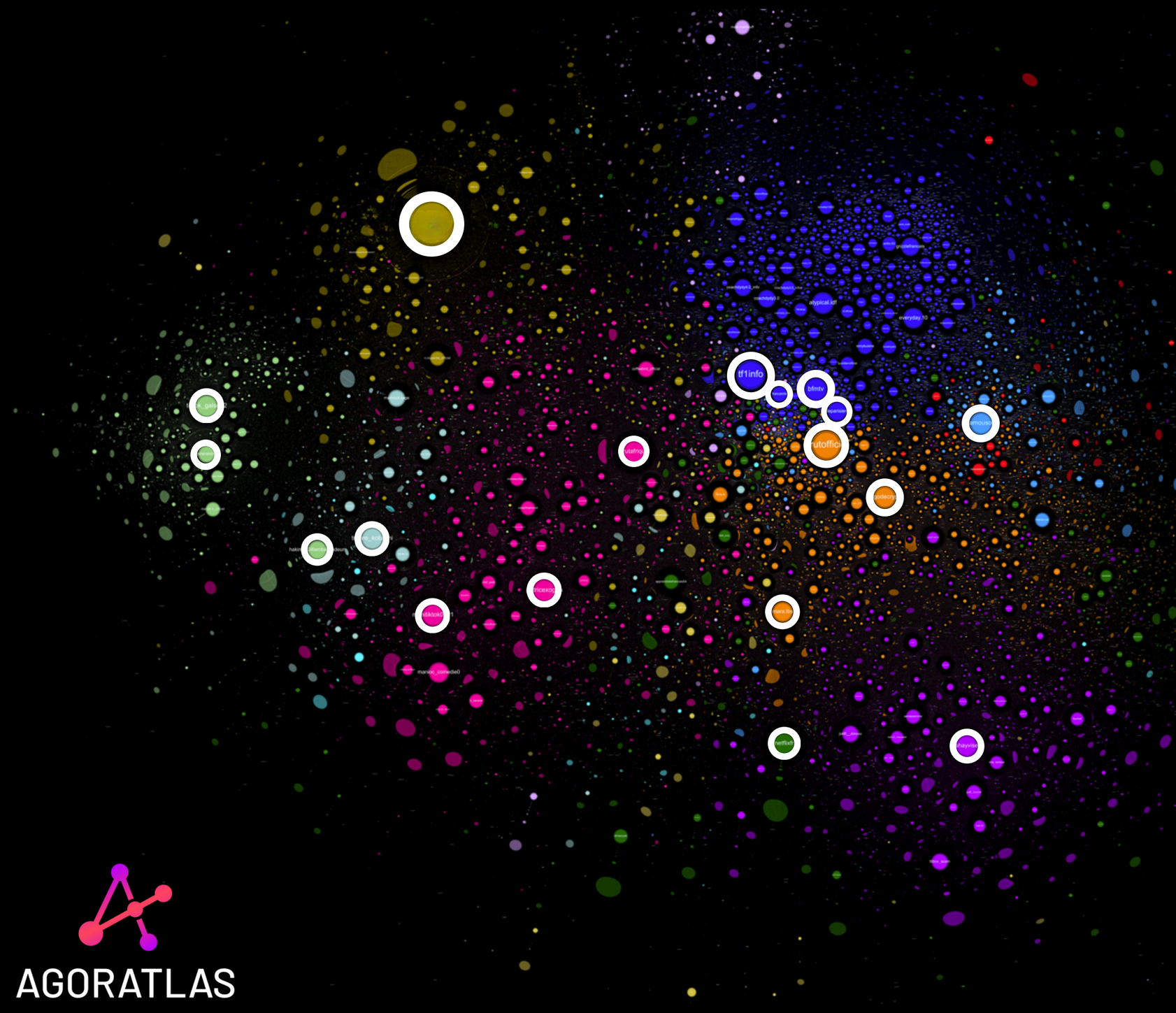
- Each attack is different
- Cannot (yet) replace humans in investigations
- Acquire **experience**, develop **tools**
- More data = better results



Cartography

One of the most powerful tools

AGORATLAS

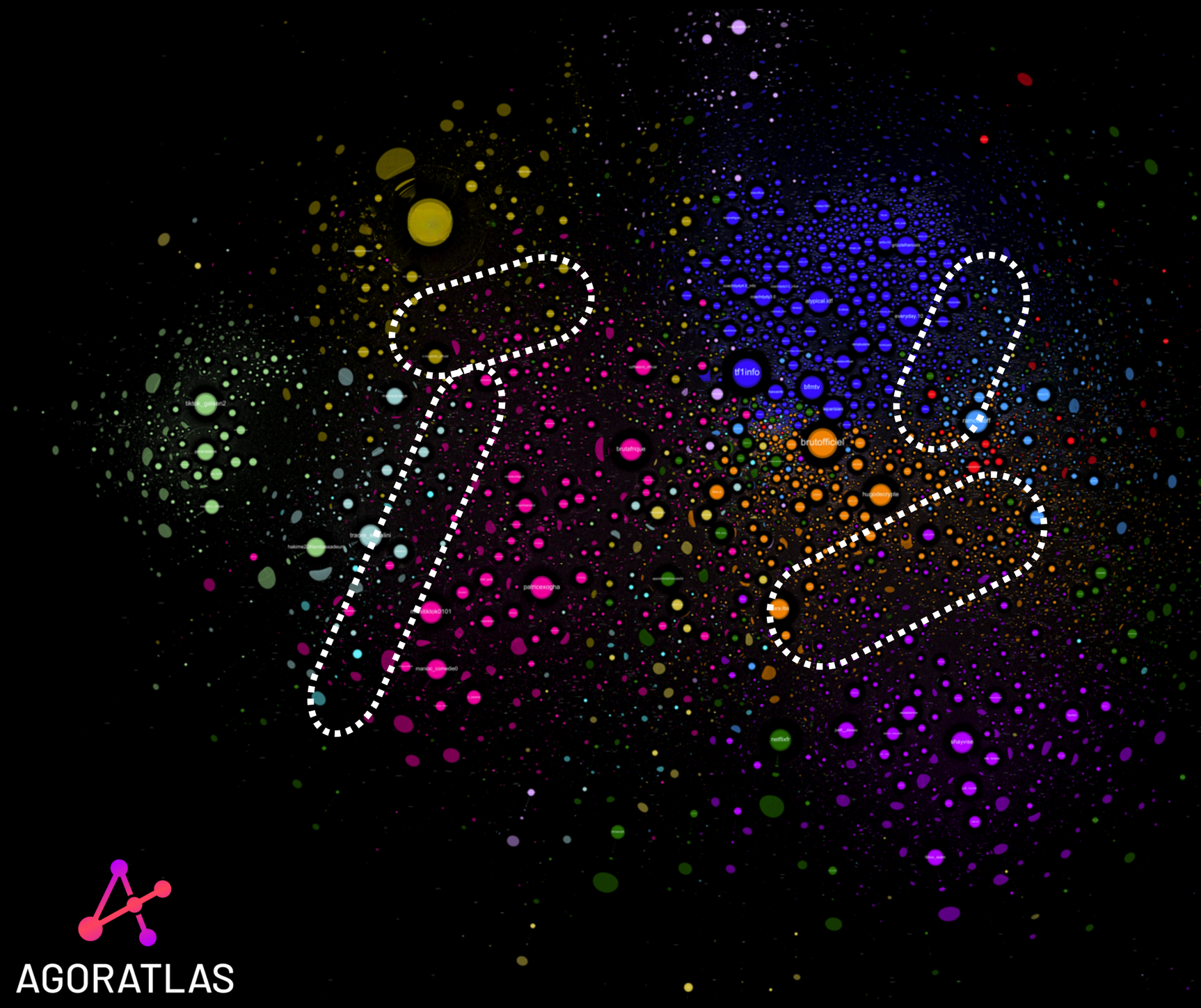


Identify
key accounts
in an instant

AGORATLAS

Identify
communities
in an instant

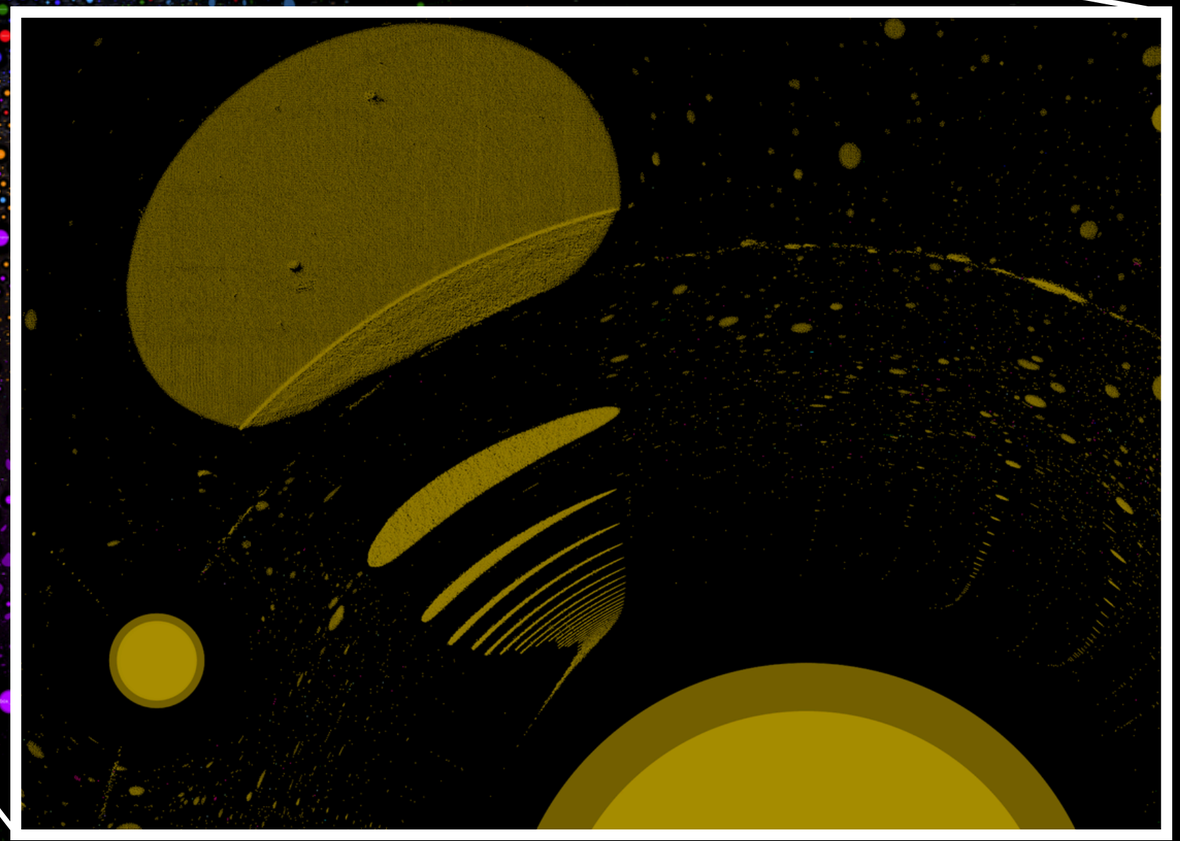
AGORATLAS



Identify
bridges
in an instant

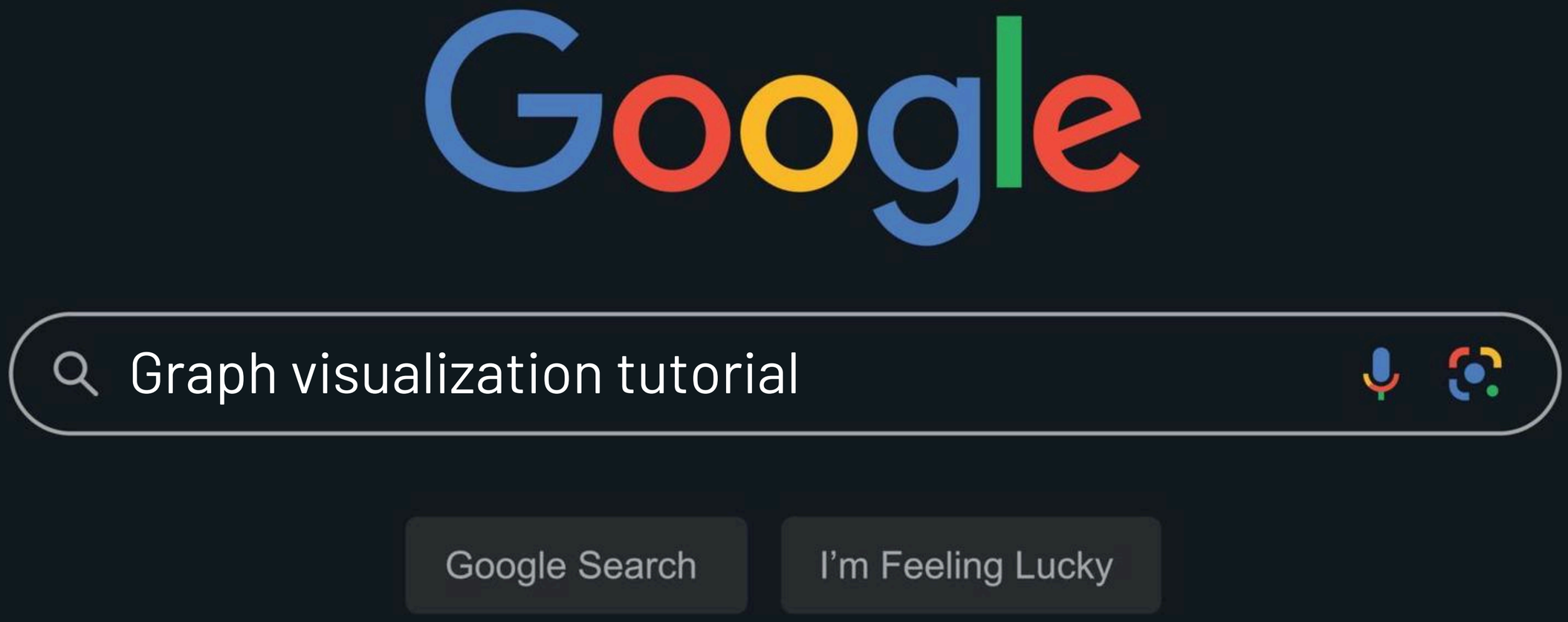
AGORATLAS

Reveal interesting
patterns and **behaviors**
at arbitrarily small scales

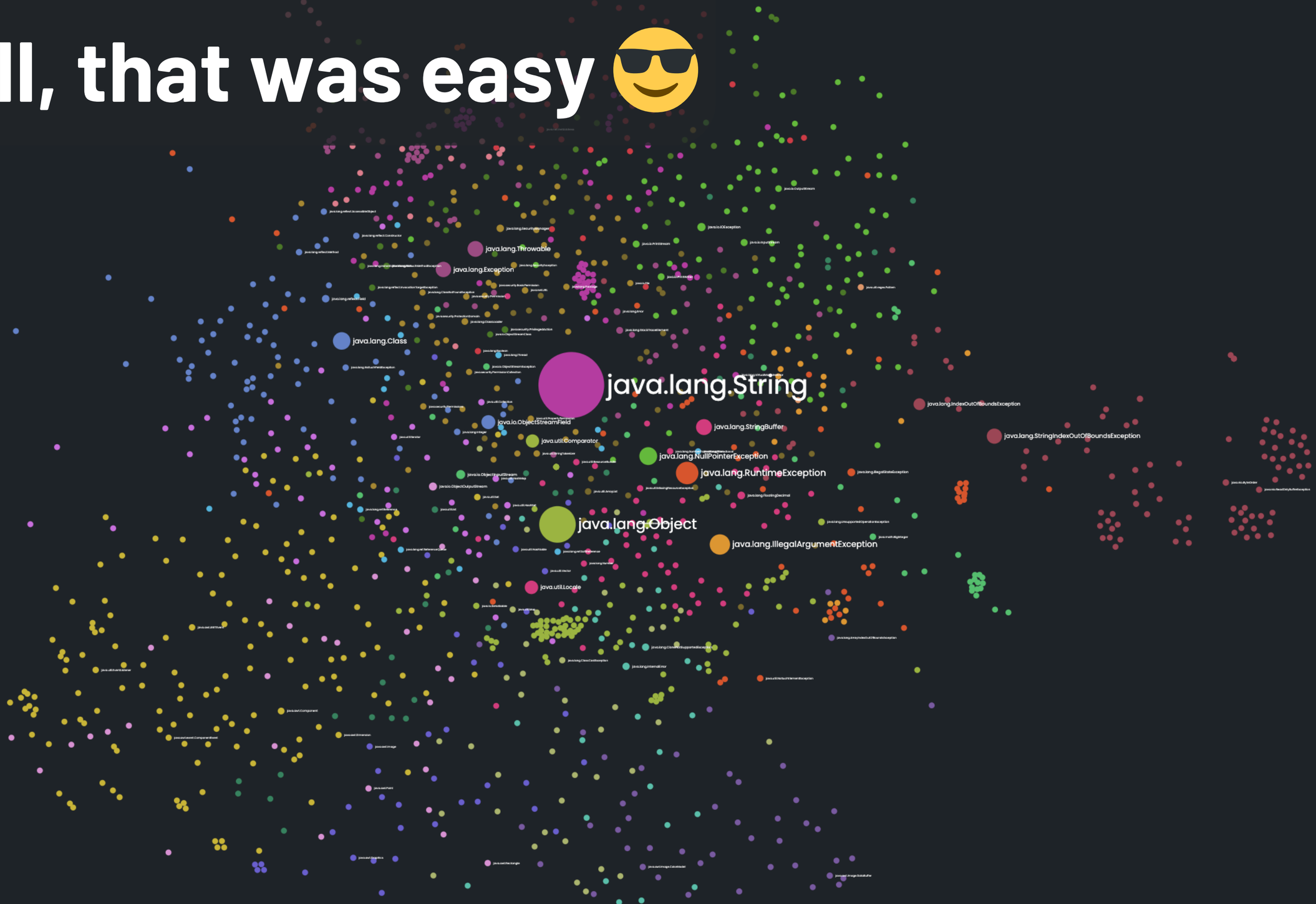


AGORATLAS

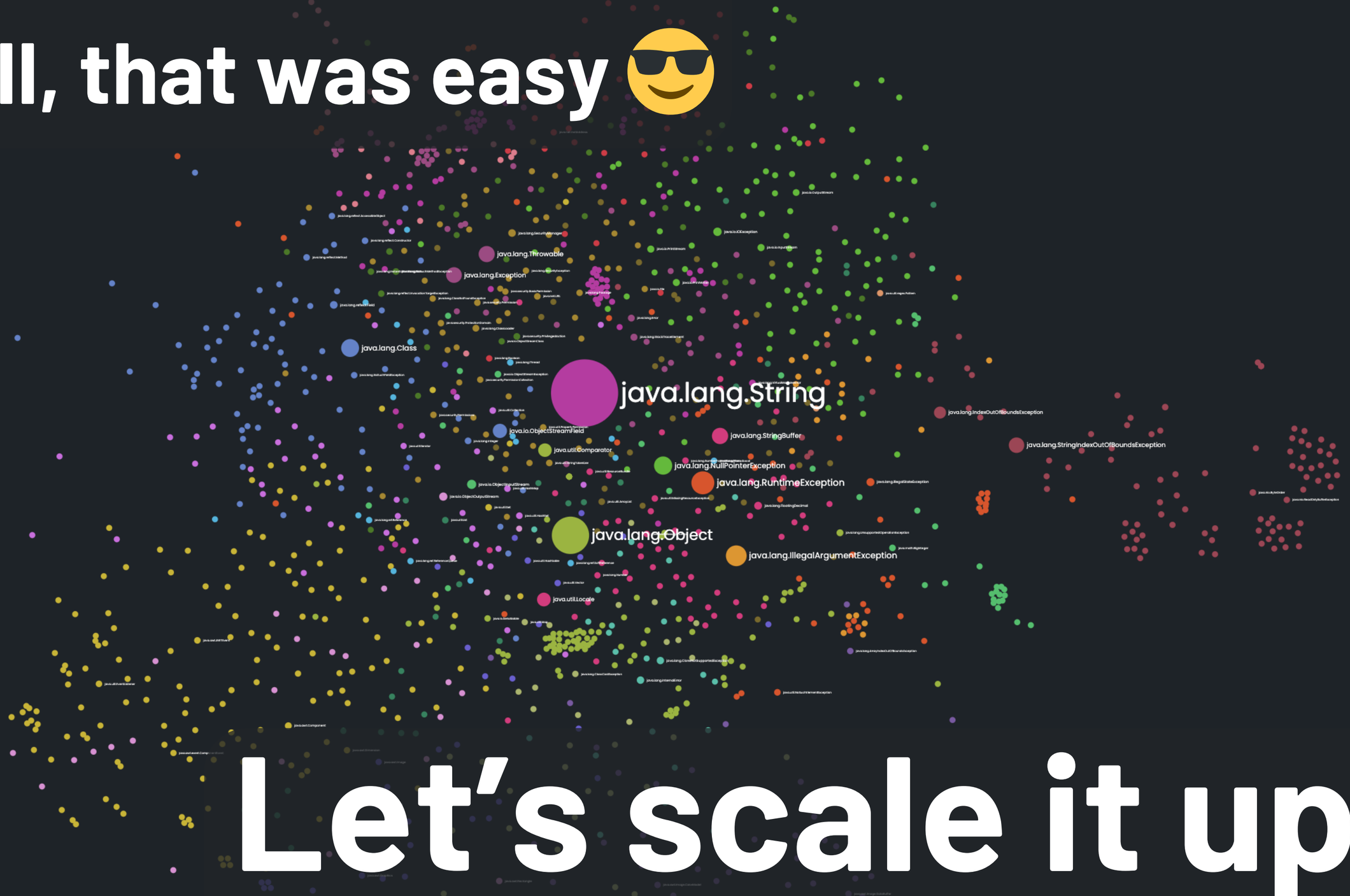
Time for a demo!



Well, that was easy 😎



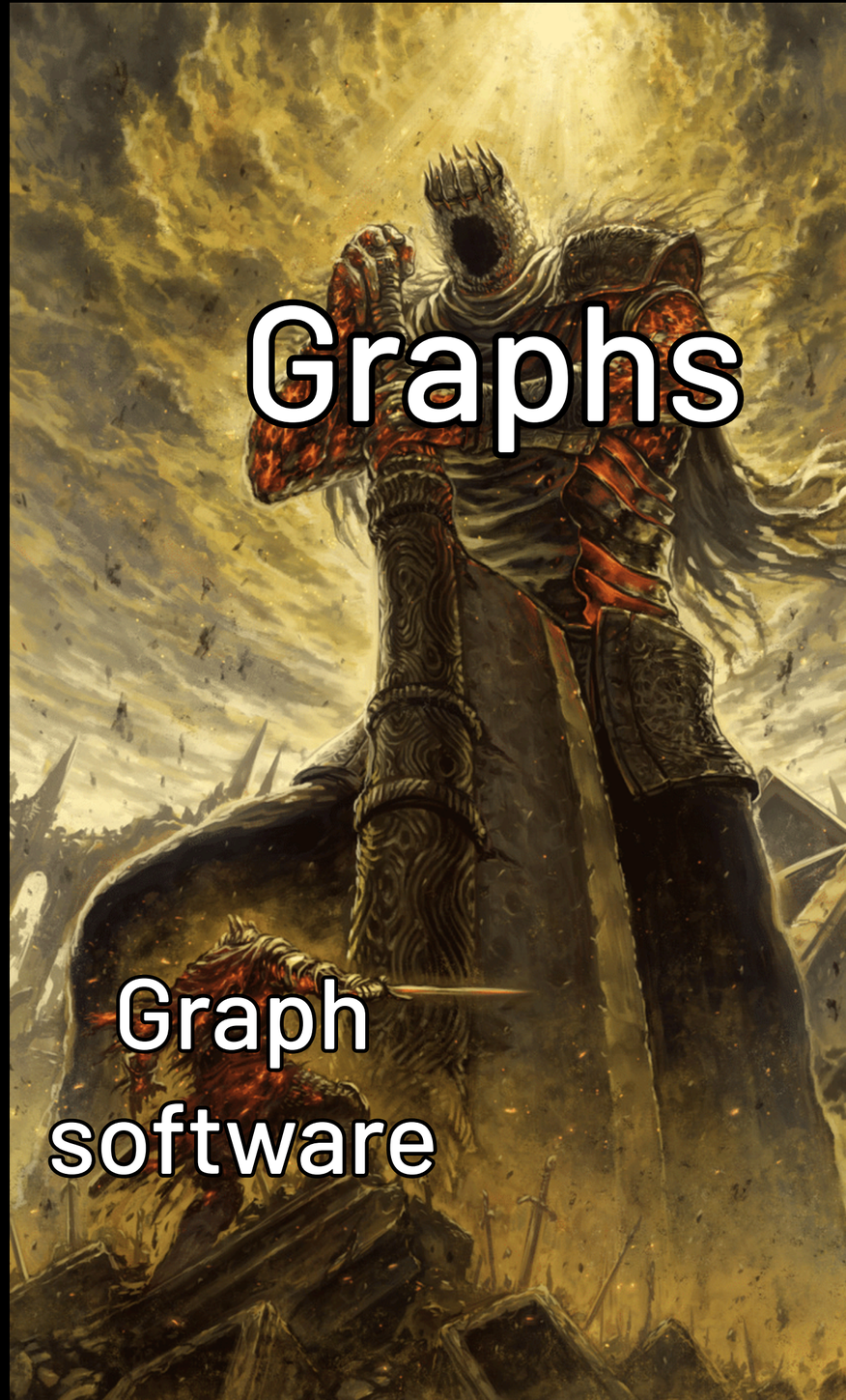
Well, that was easy 😎



Let’s scale it up!

Things start breaking very quickly!

- Algorithms become slow
- High memory usage
- Random crashes



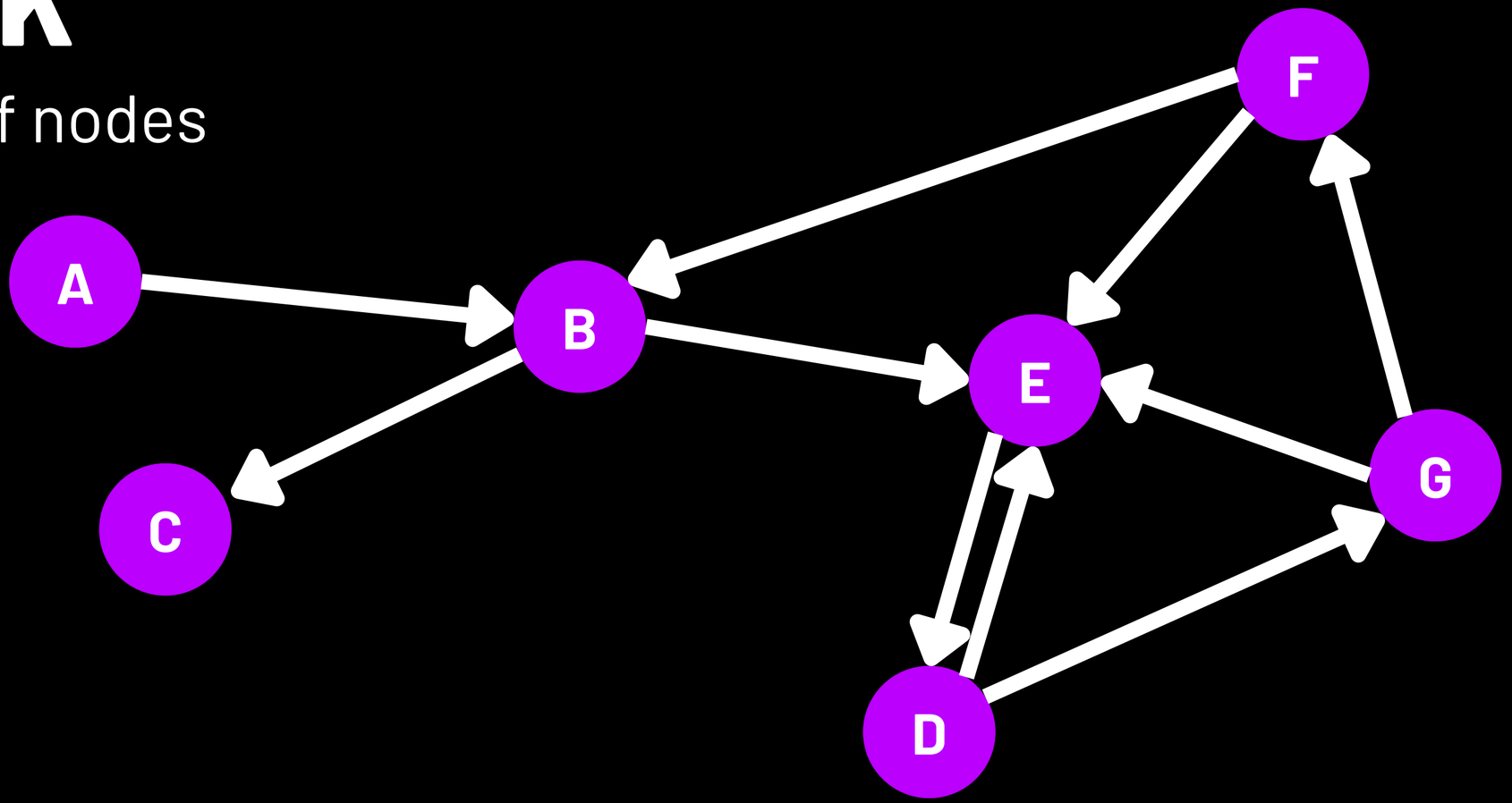
Optimizing algorithms

3 main algorithms for graph cartography:

- **PageRank** for the importance of nodes
- **Louvain** to split into communities
- **ForceAtlas2** for the 2D layout

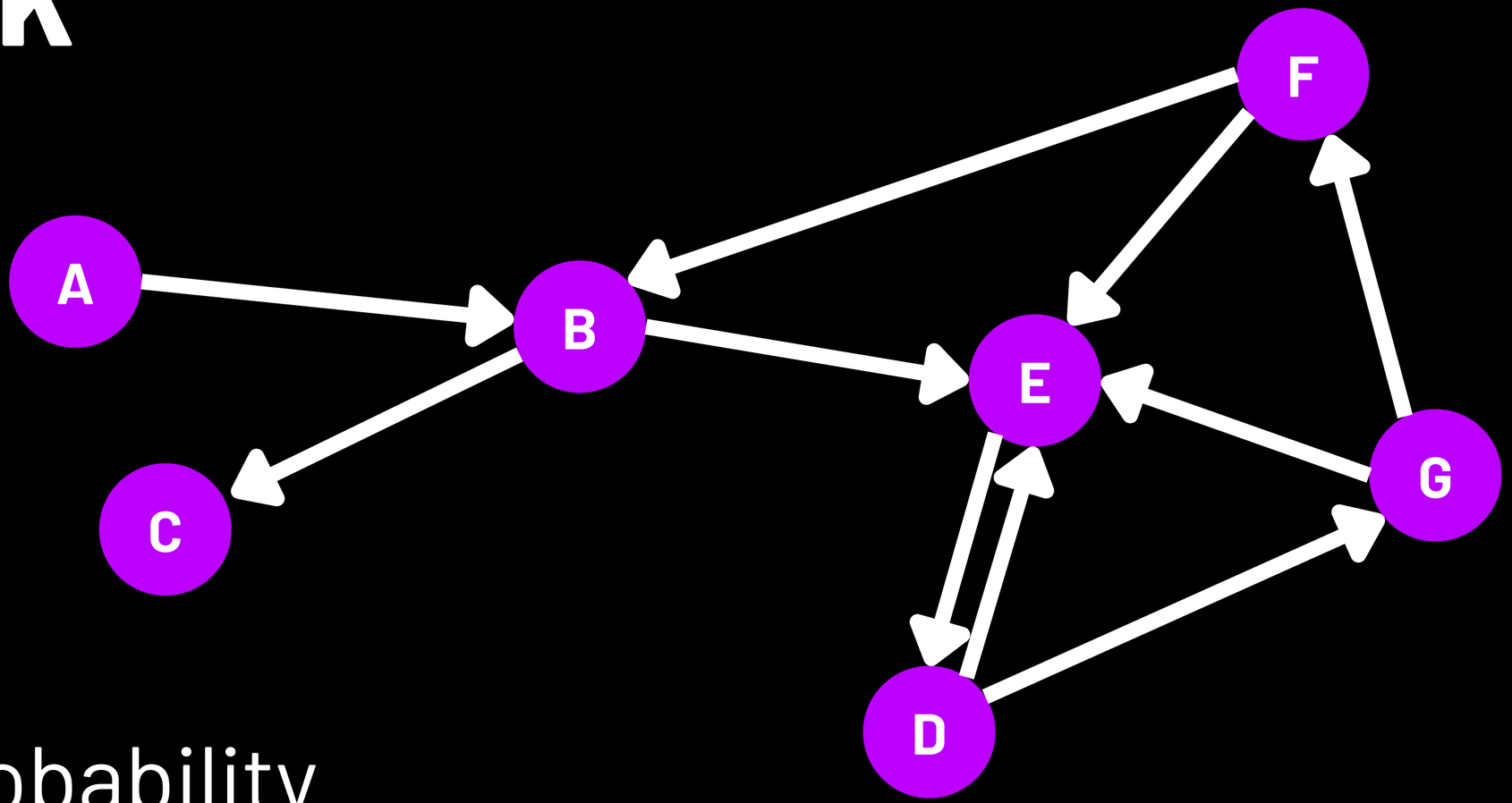
PageRank

for the importance of nodes



AGORATLAS

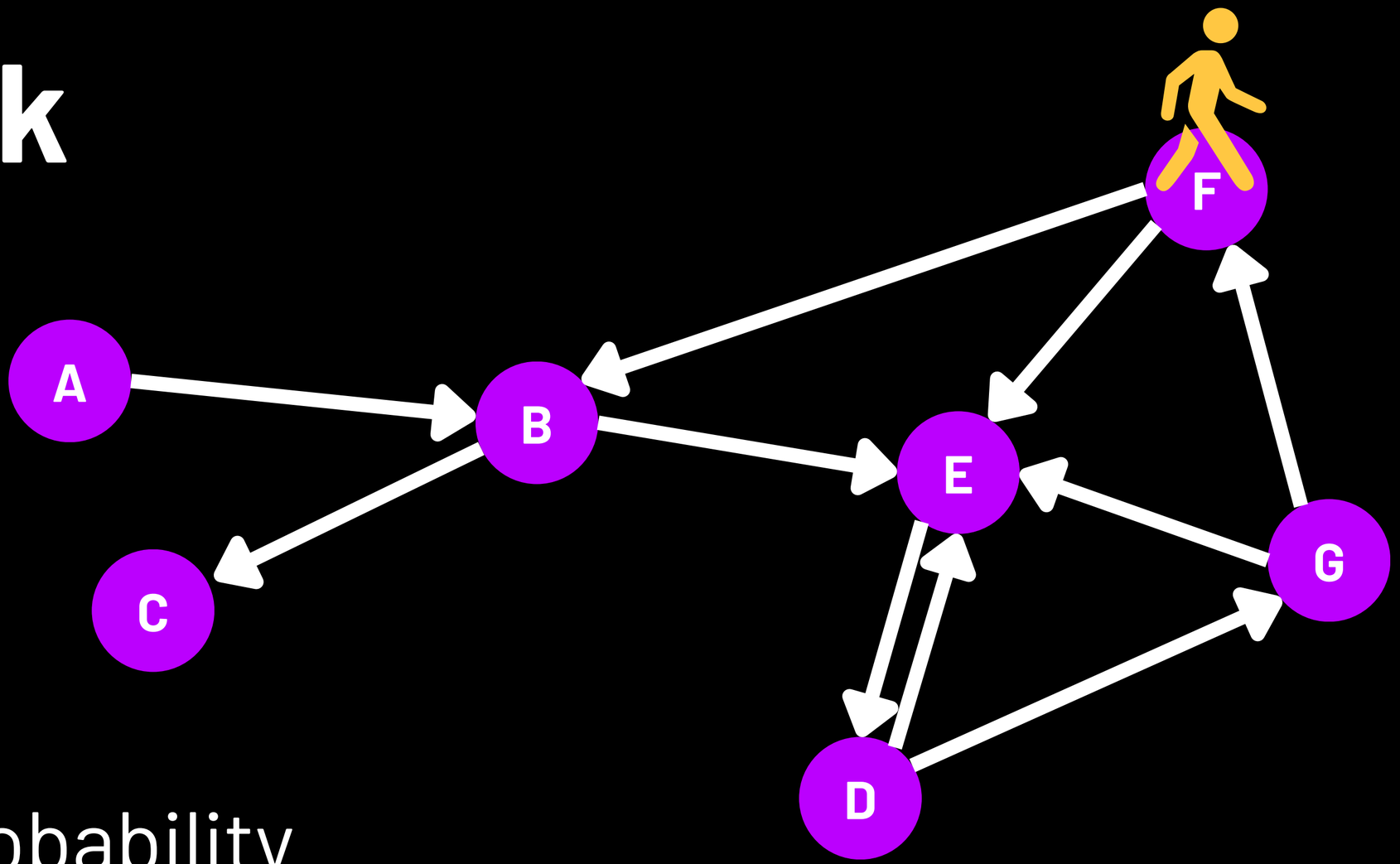
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

AGORATLAS

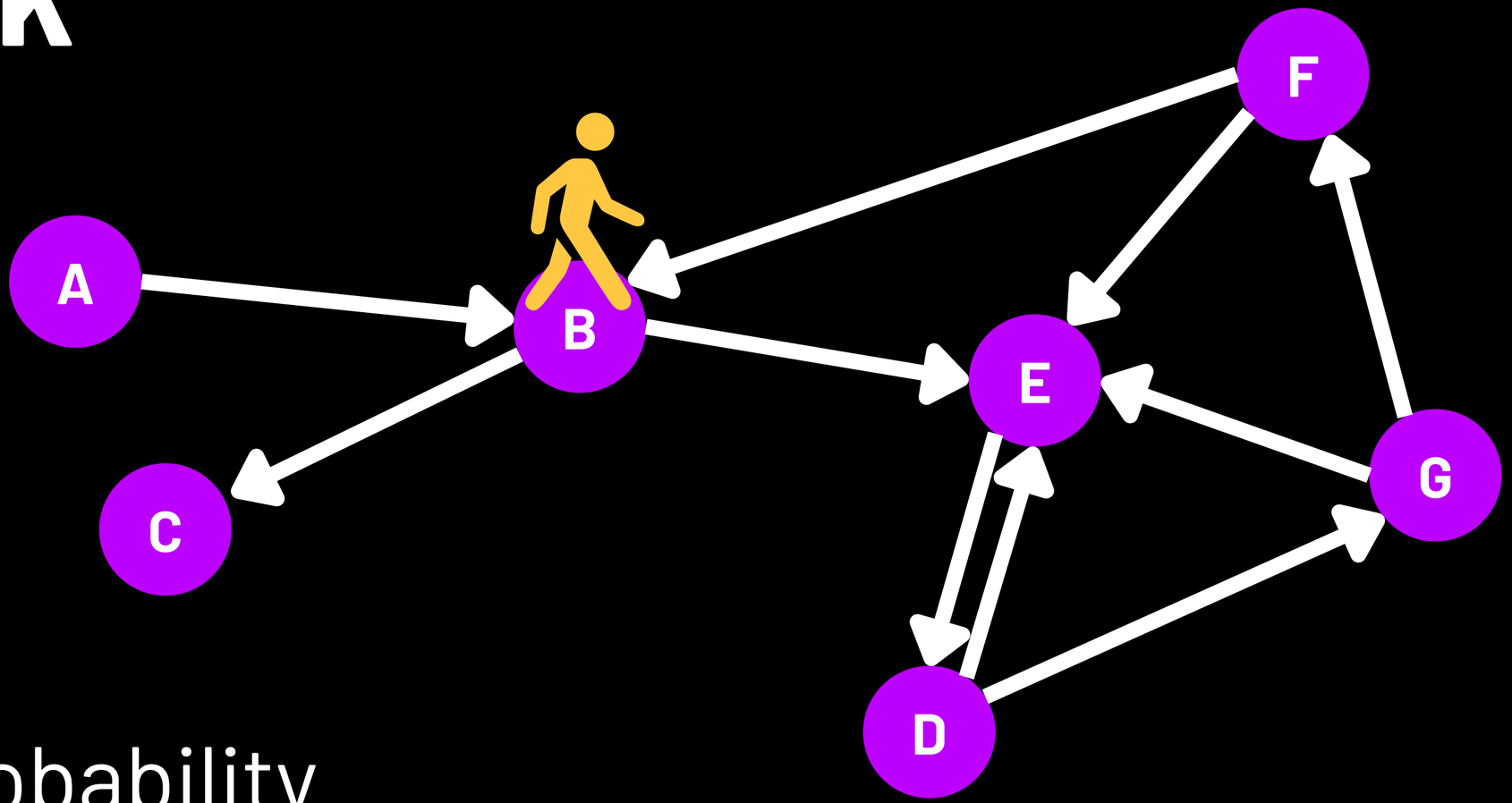
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=0

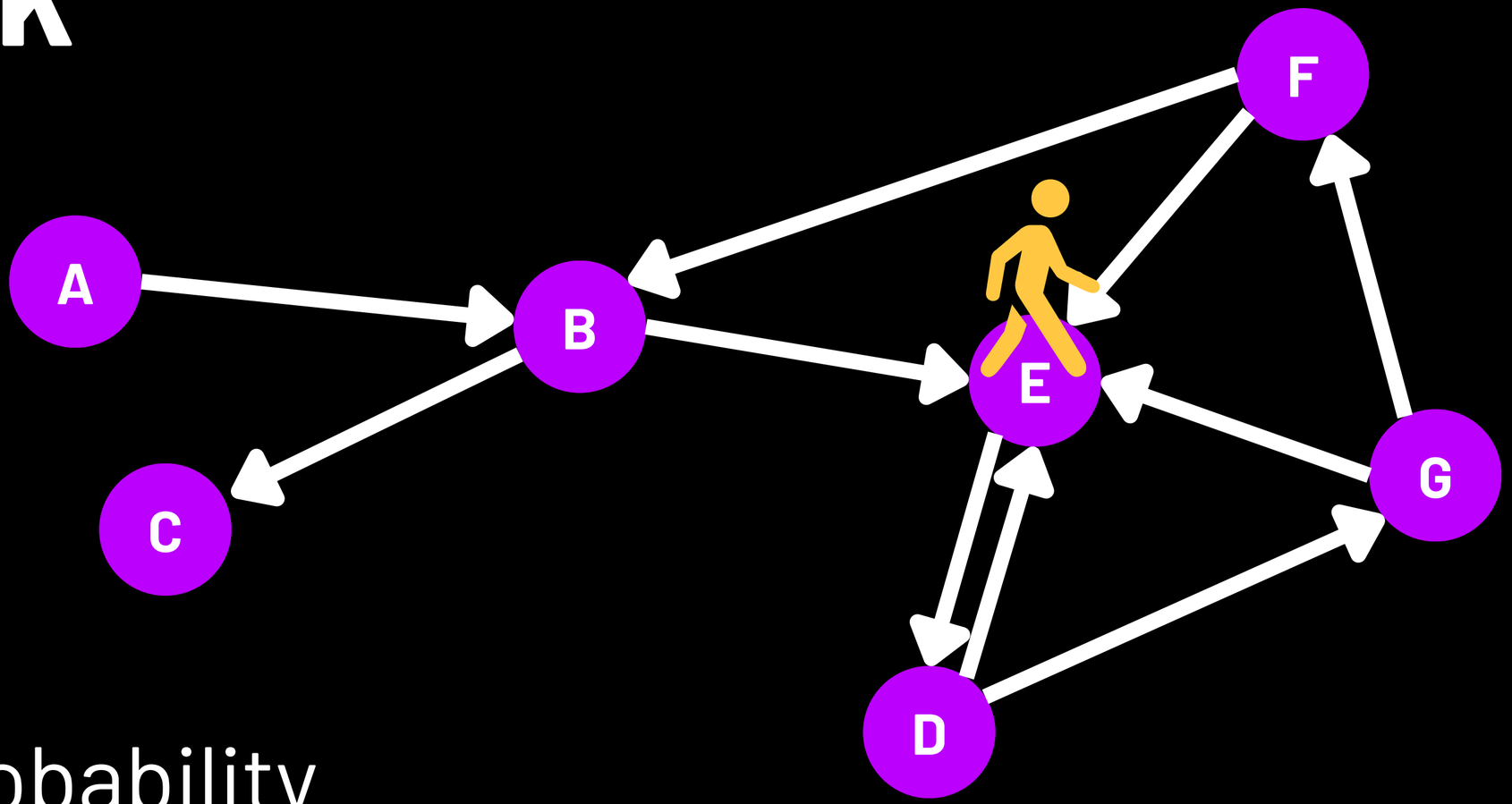
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=1

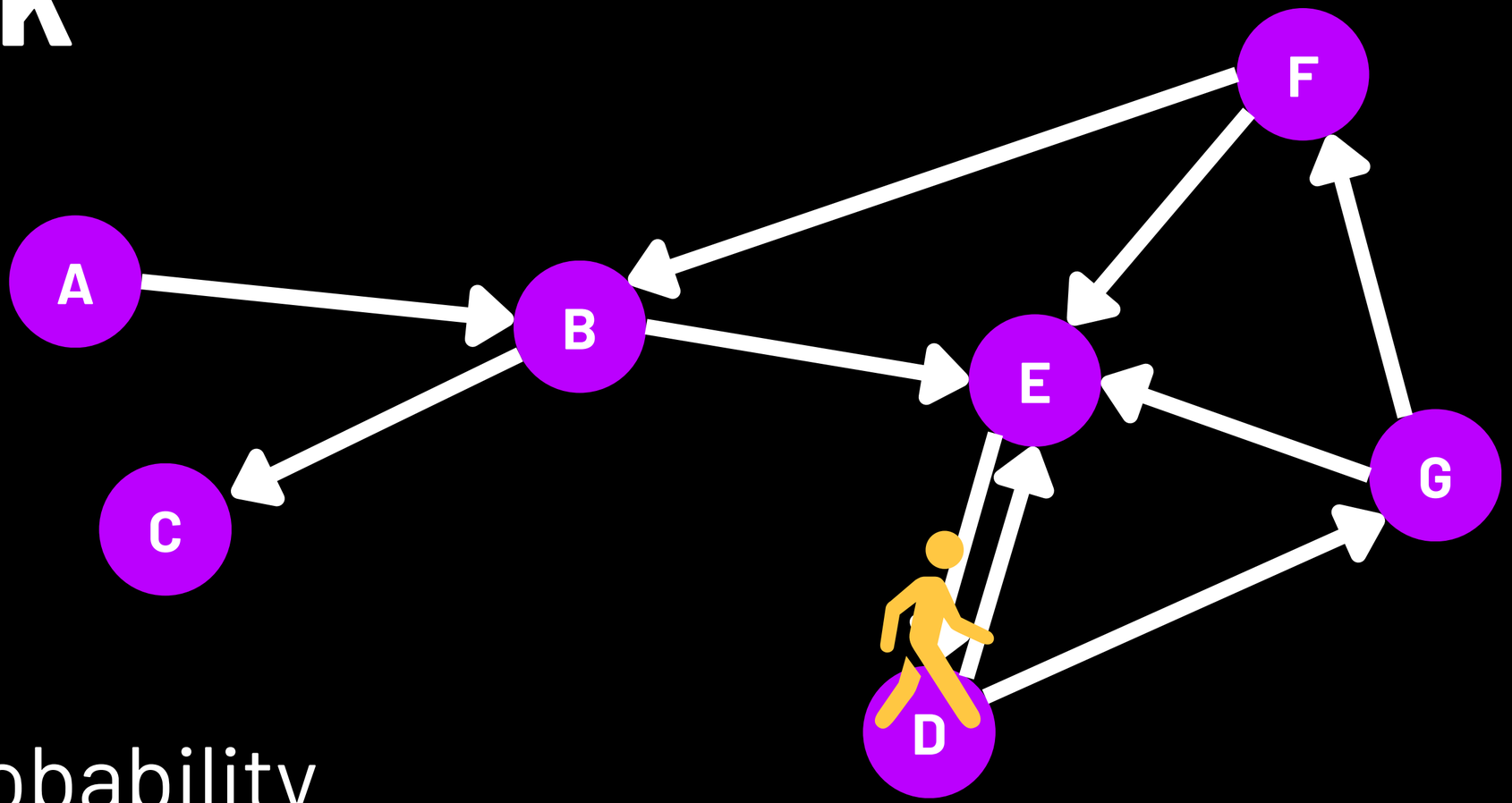
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=2

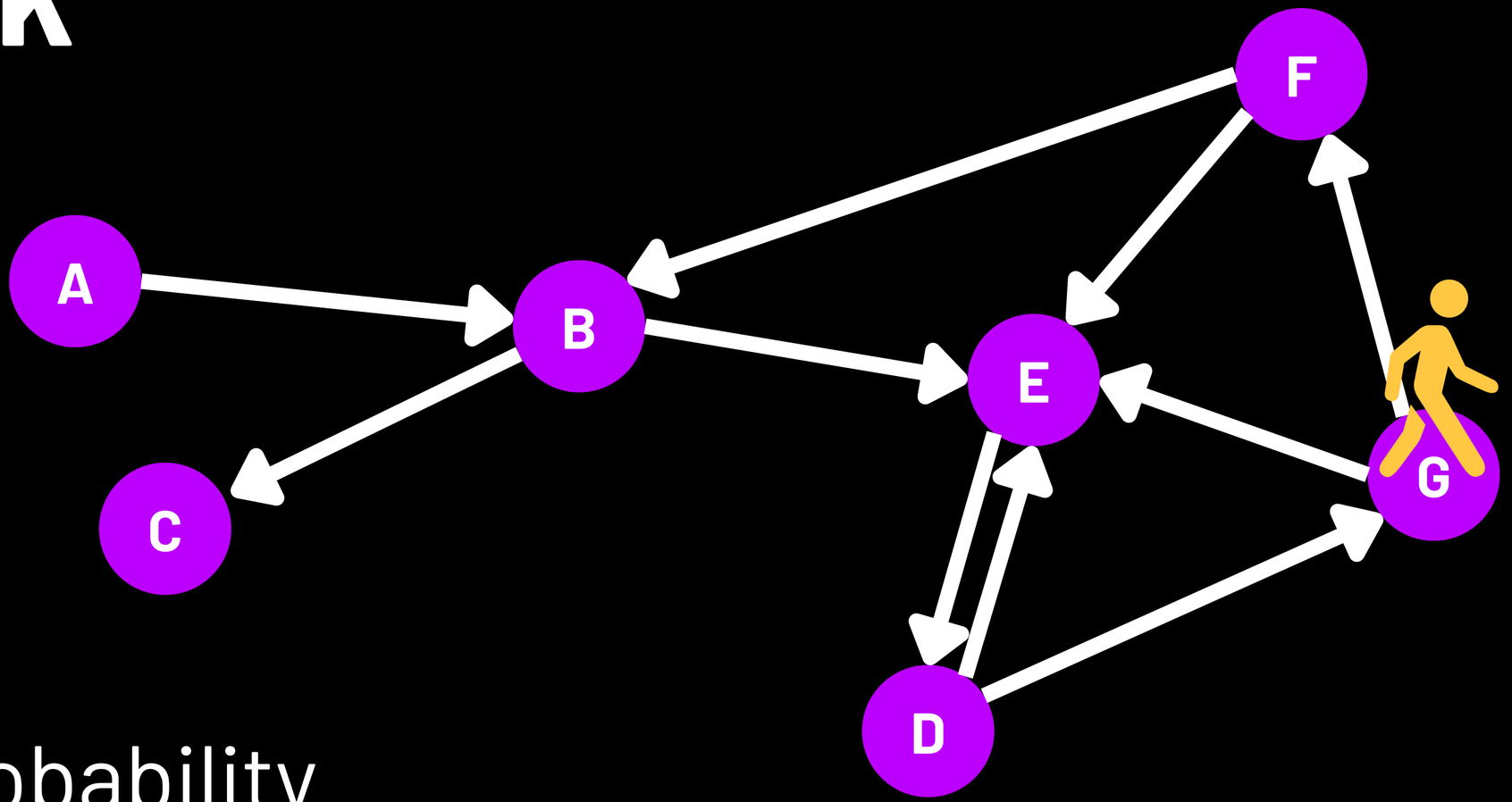
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=3

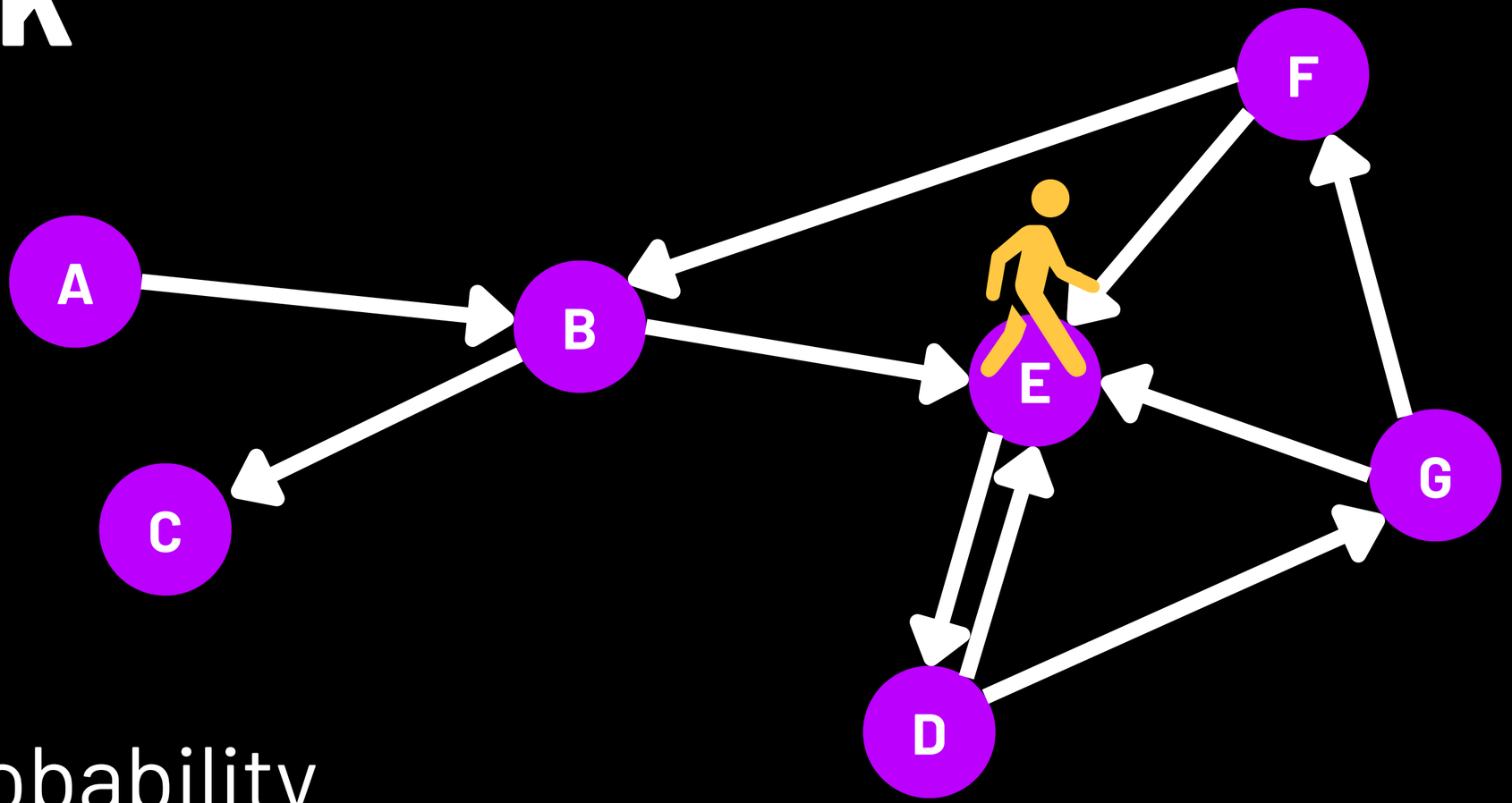
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=4

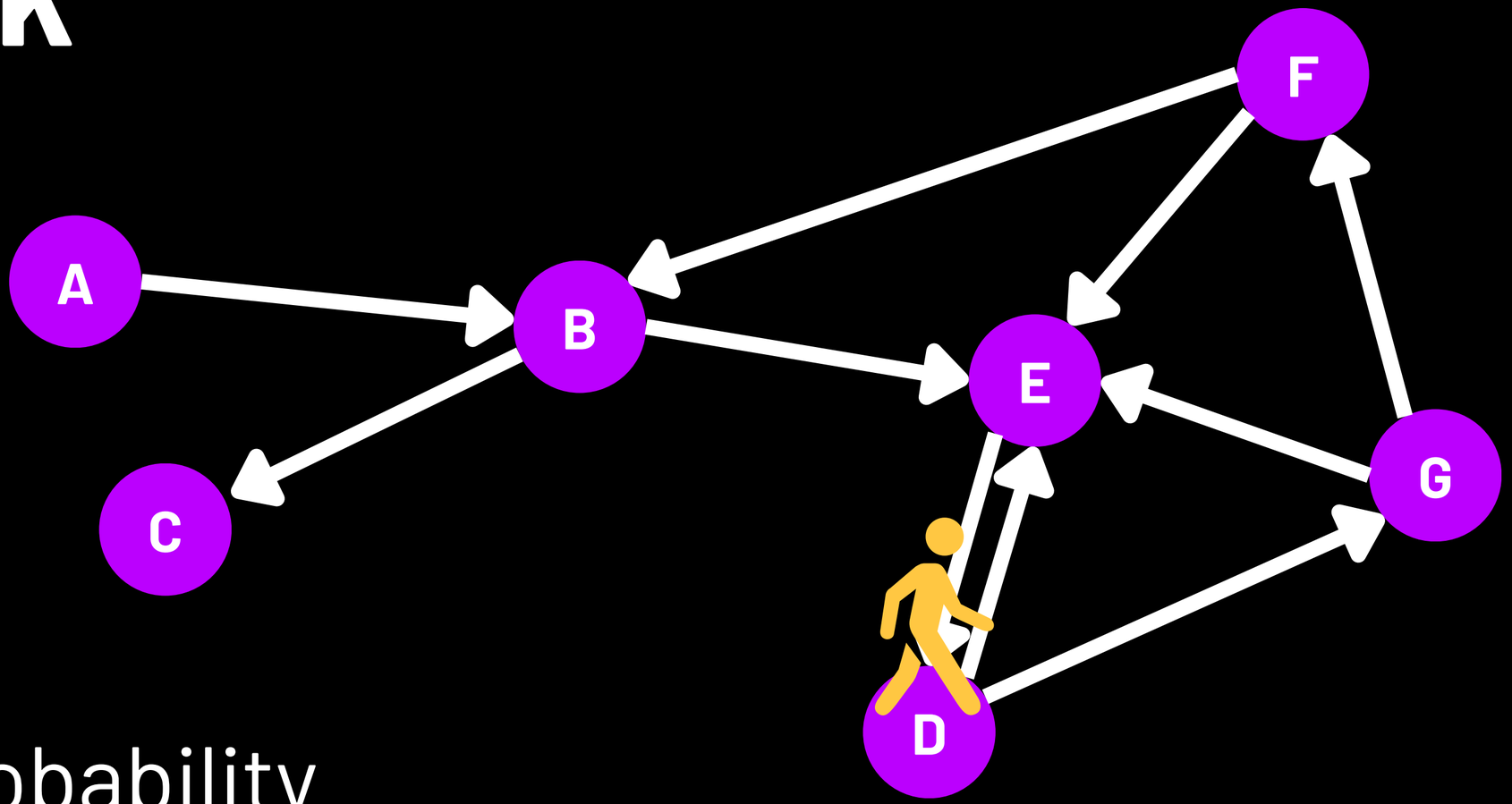
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=5

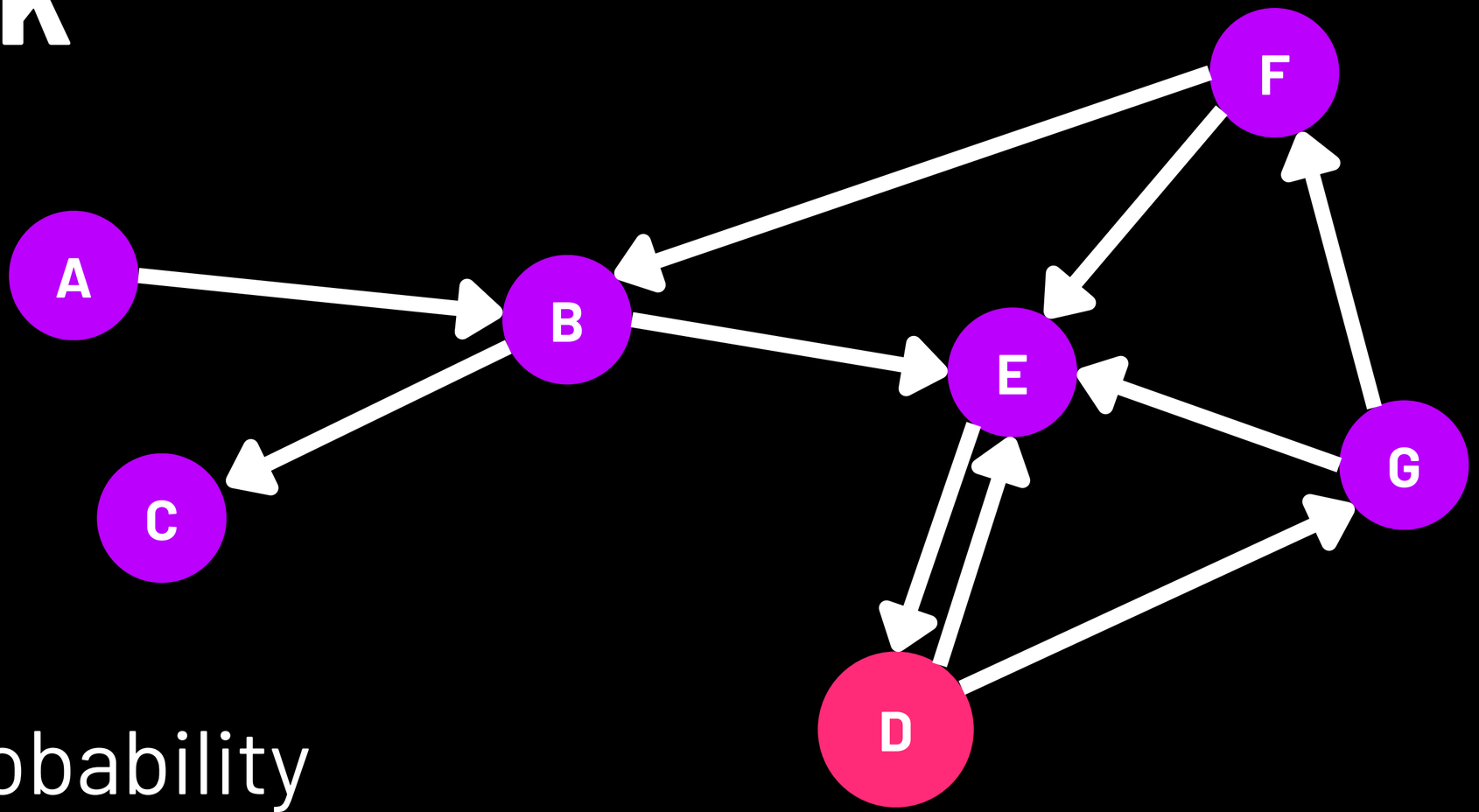
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

K=6

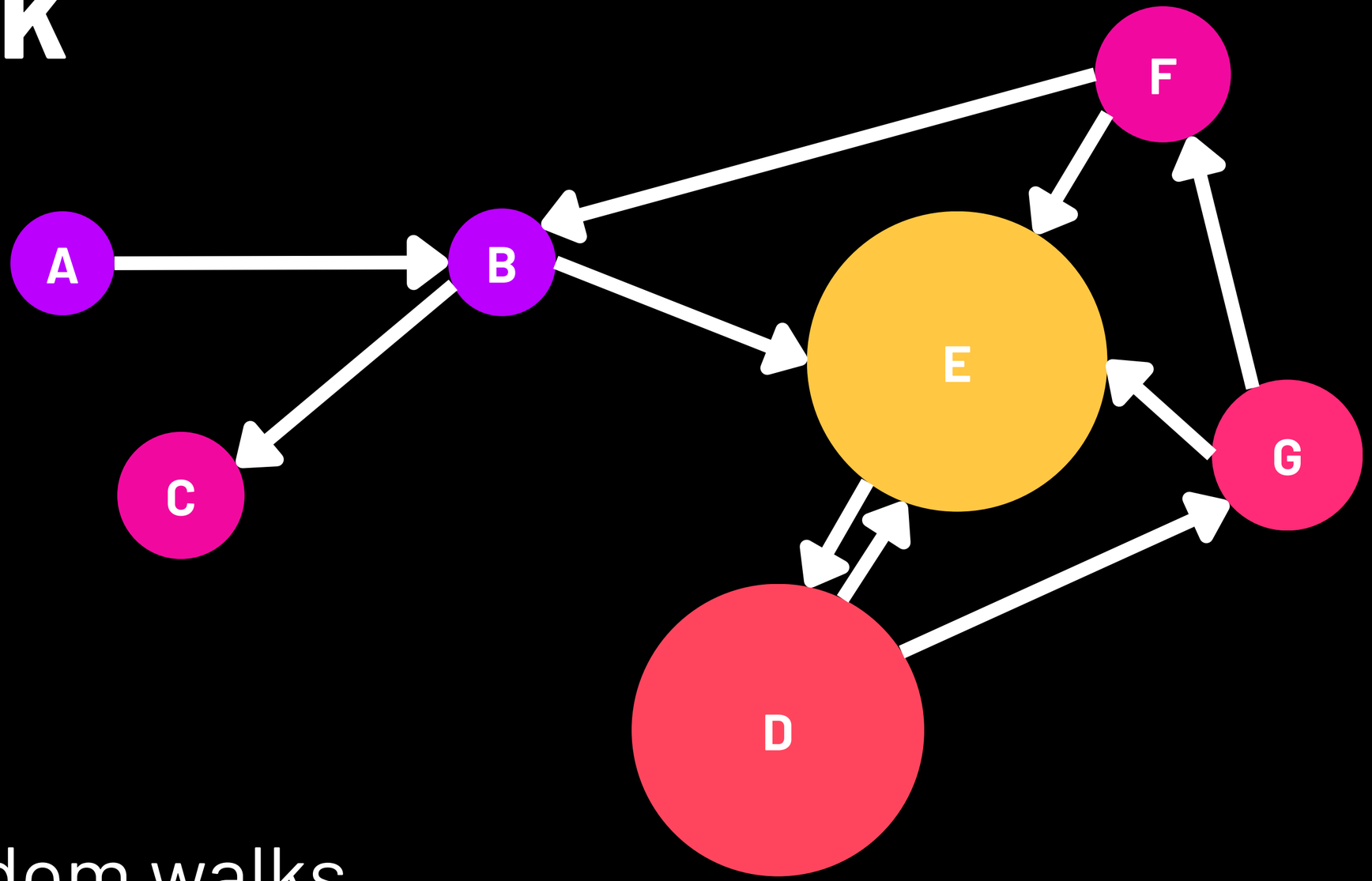
PageRank



“What is the probability
that a **random walk**
starting from a random node
will **visit node N** after K steps?”

AGORATLAS

PageRank

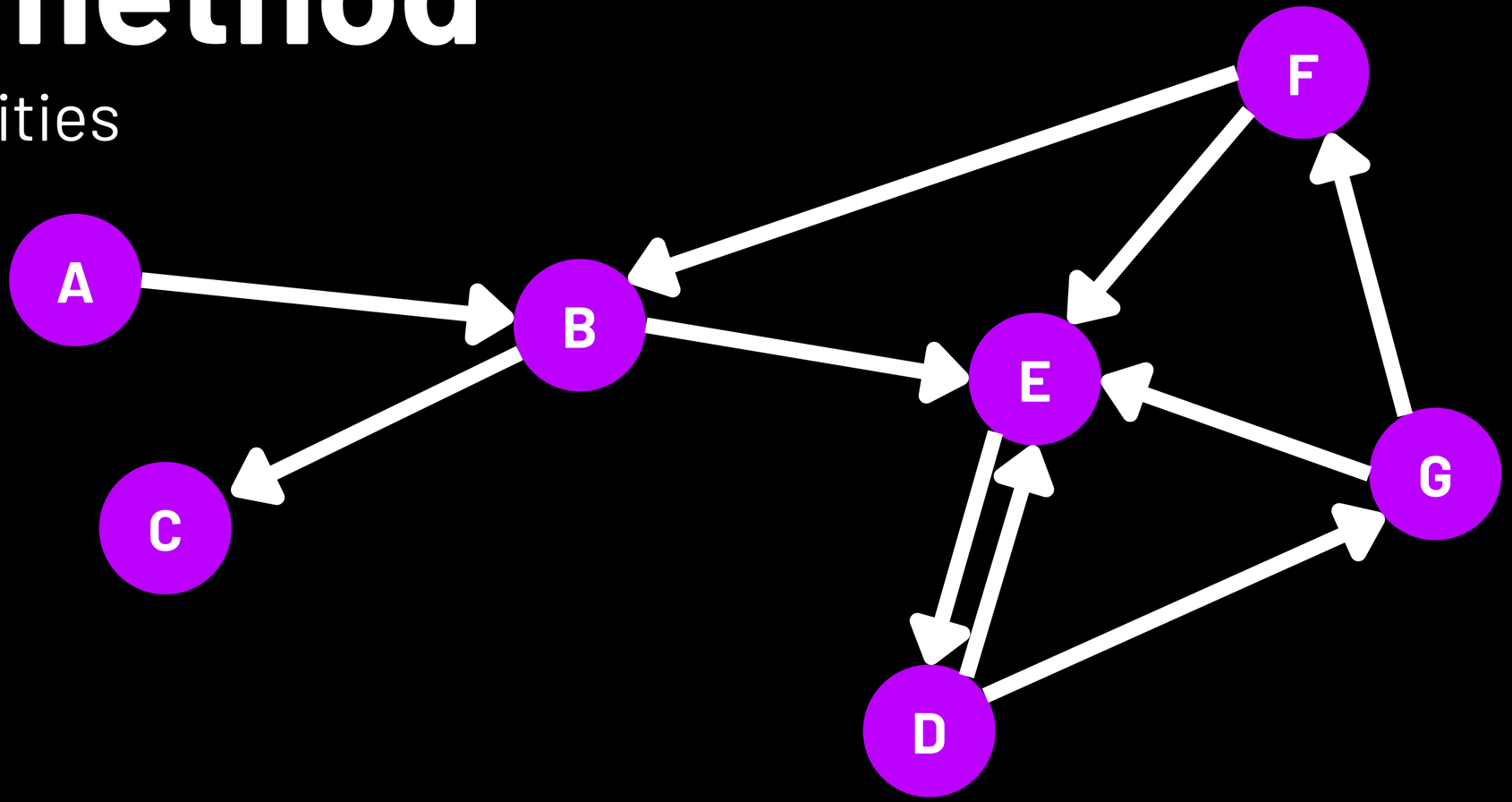


After many random walks...

AGORATLAS

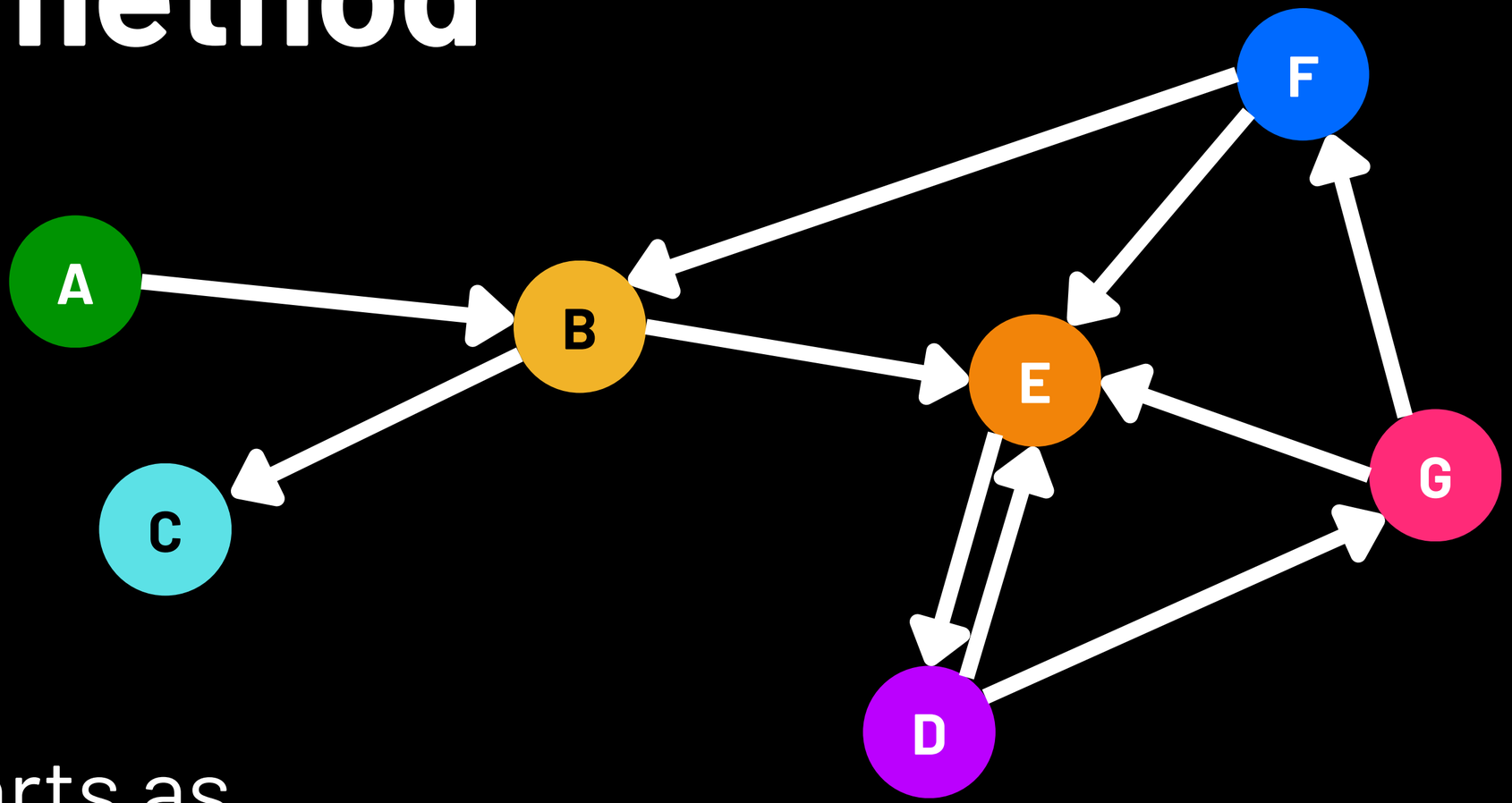
Louvain method

to split into communities



AGORATLAS

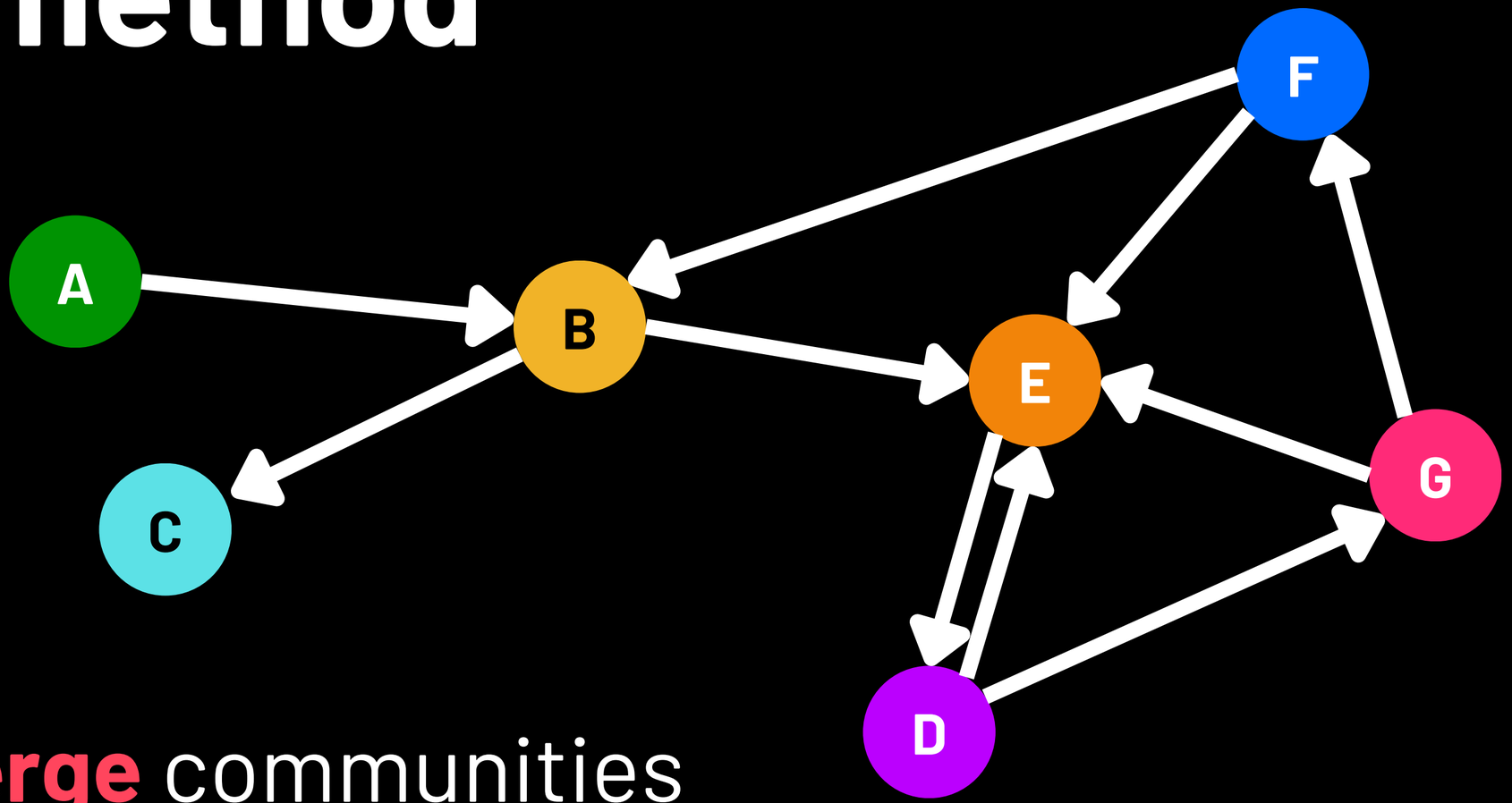
Louvain method



Every node starts as
its own community

AGORATLAS

Louvain method



Then, try to **merge** communities

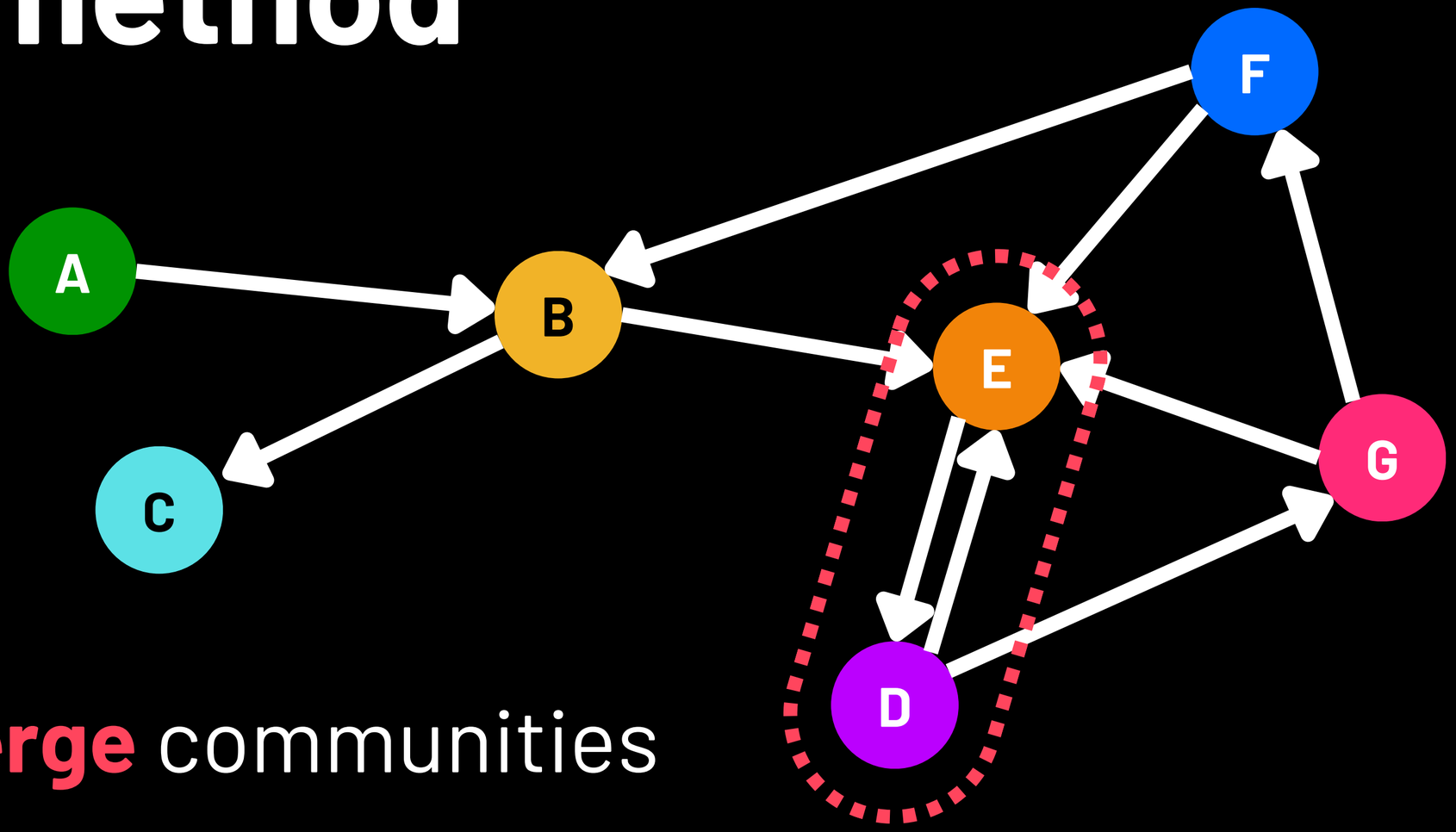
Objective:

- Maximize edges within groups
- Minimize edges between groups

} **“modularity”**

AGORATLAS

Louvain method

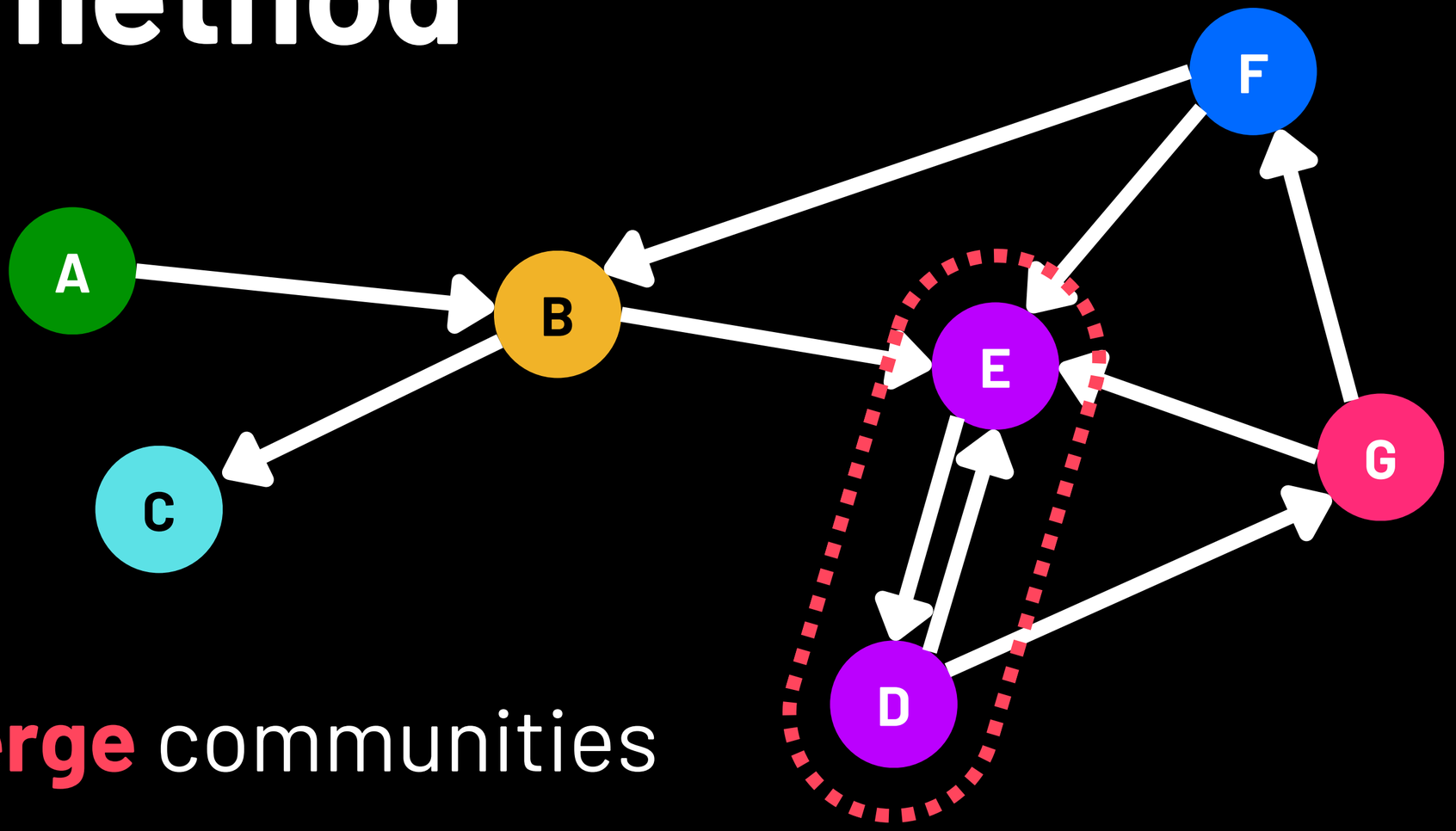


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

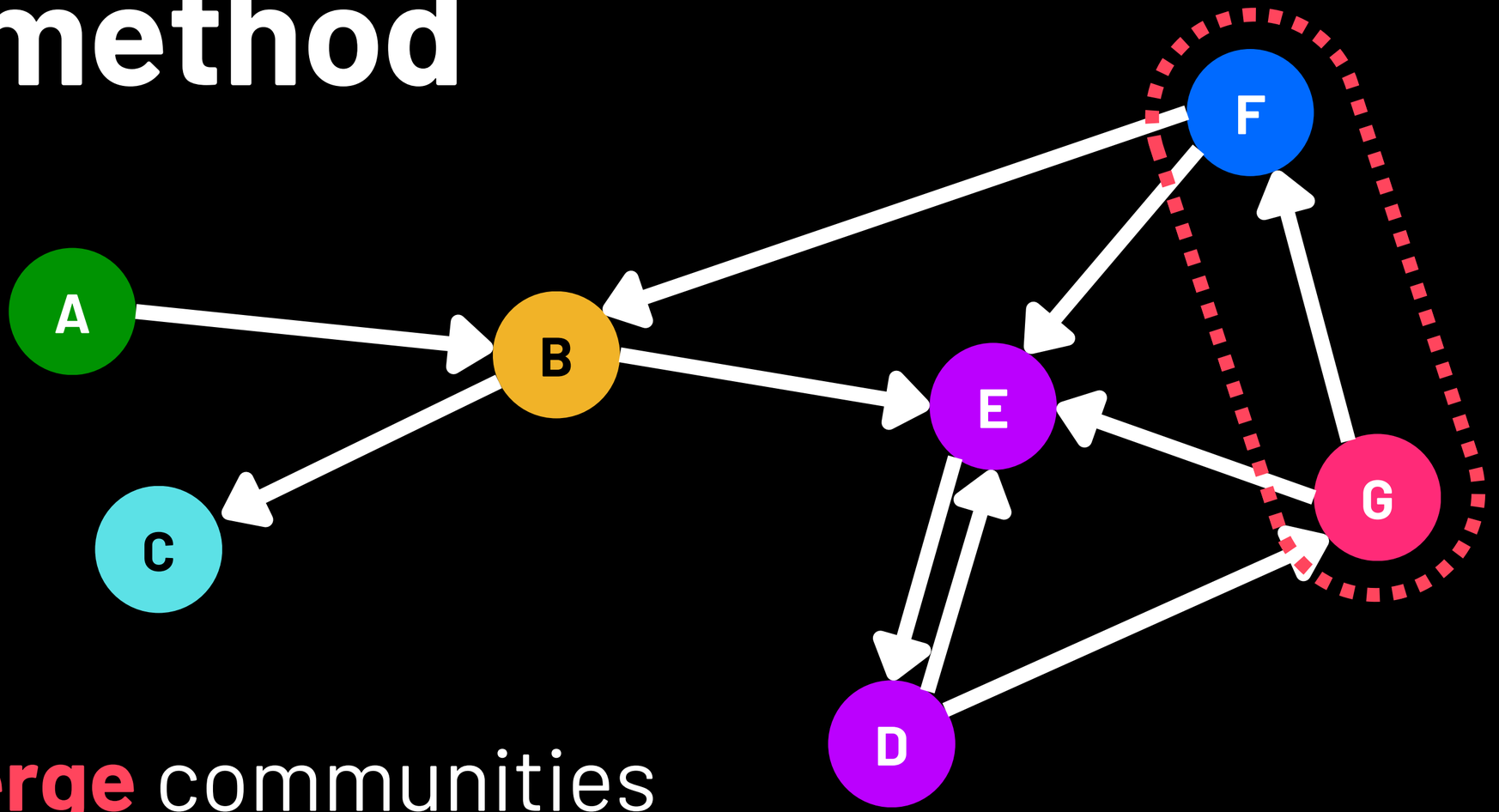


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

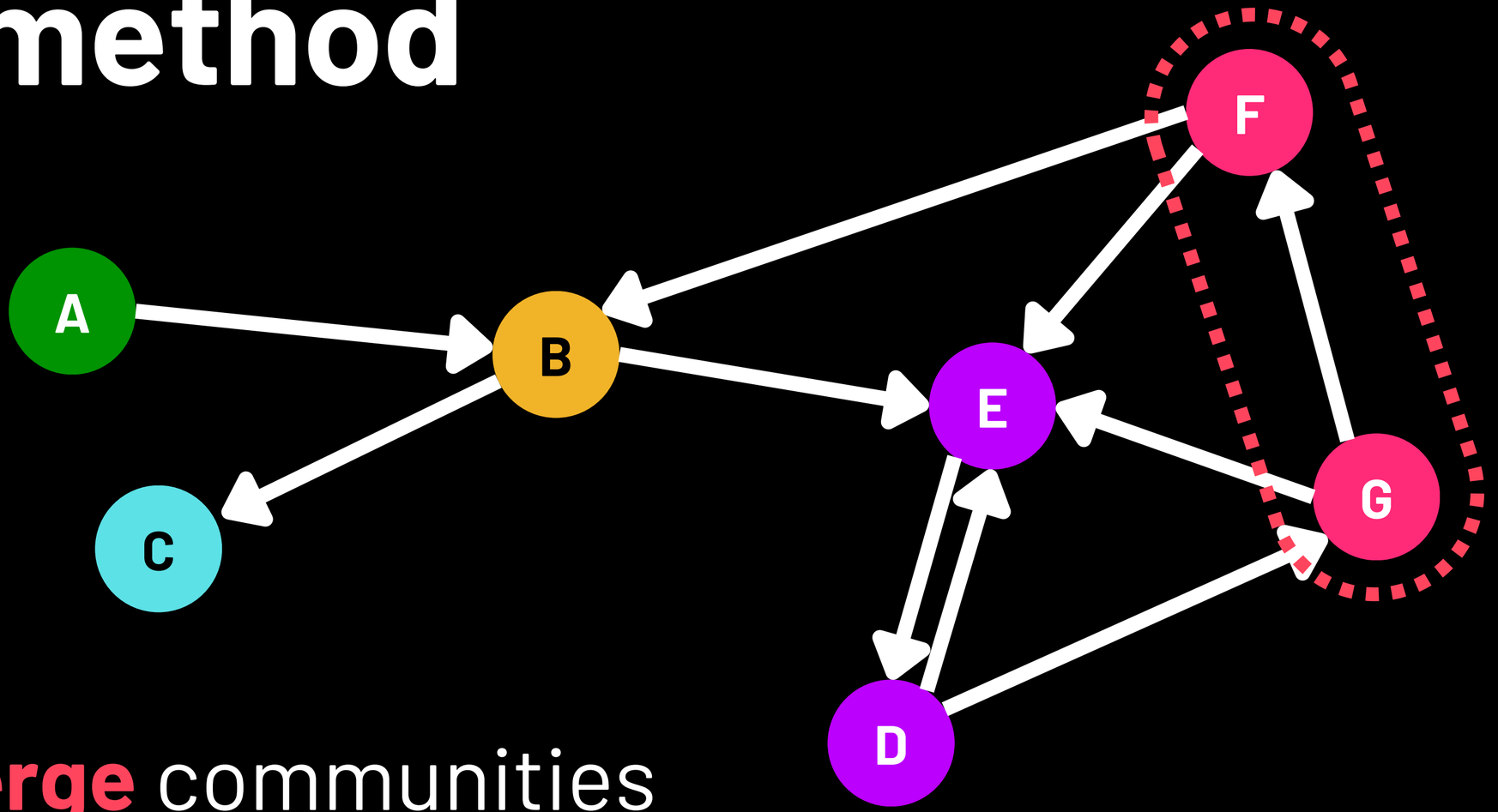


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

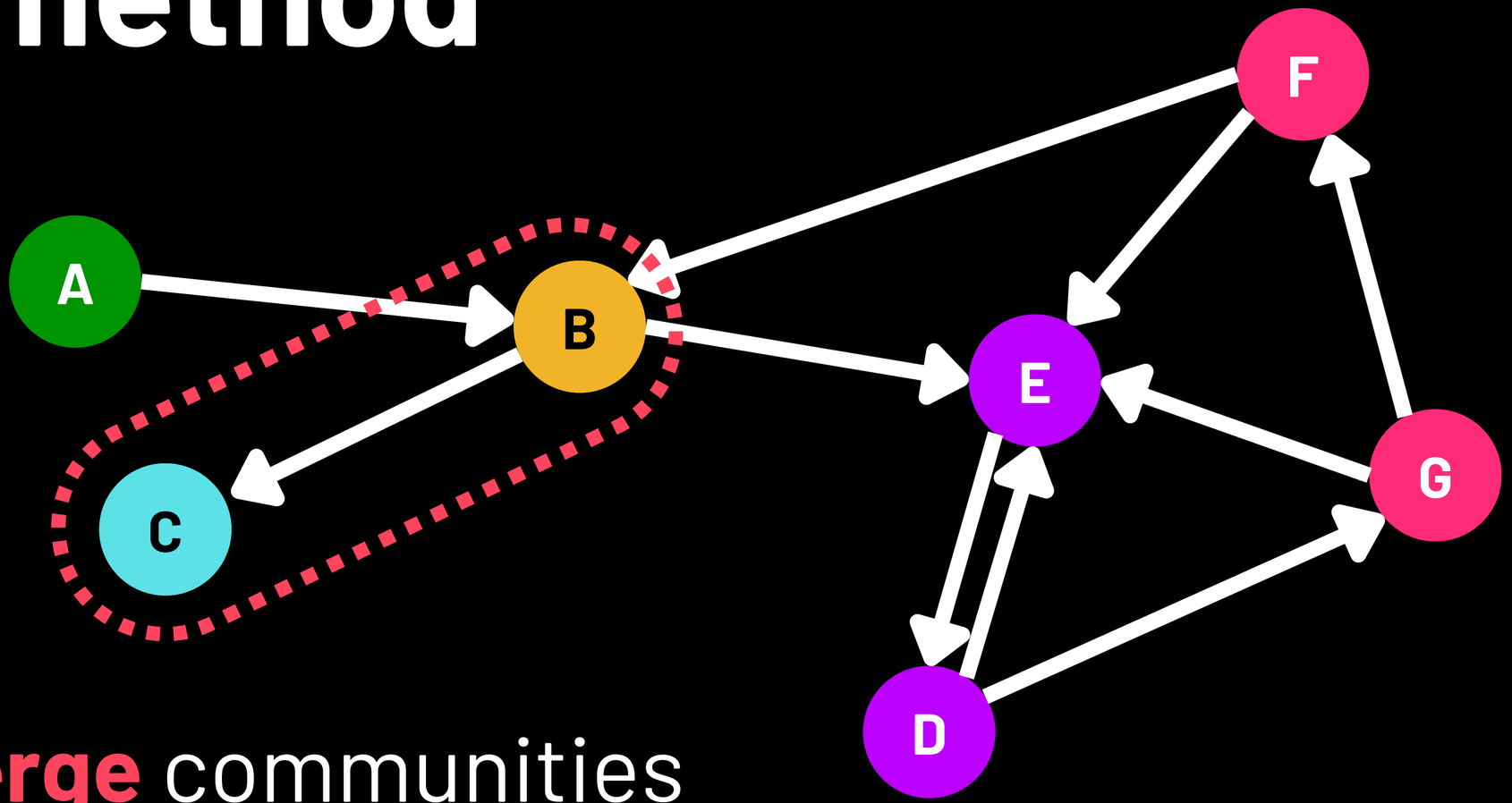


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

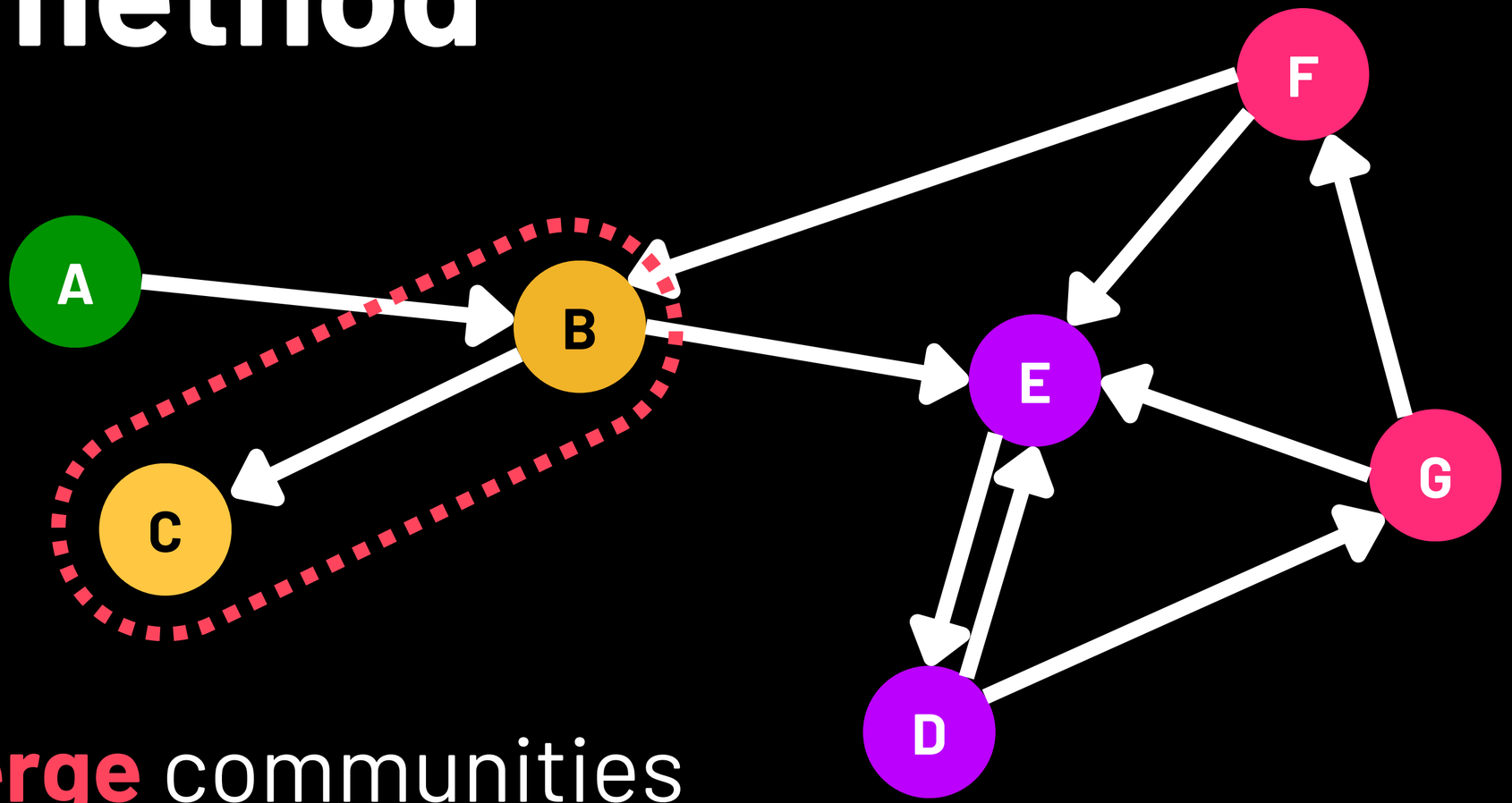


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

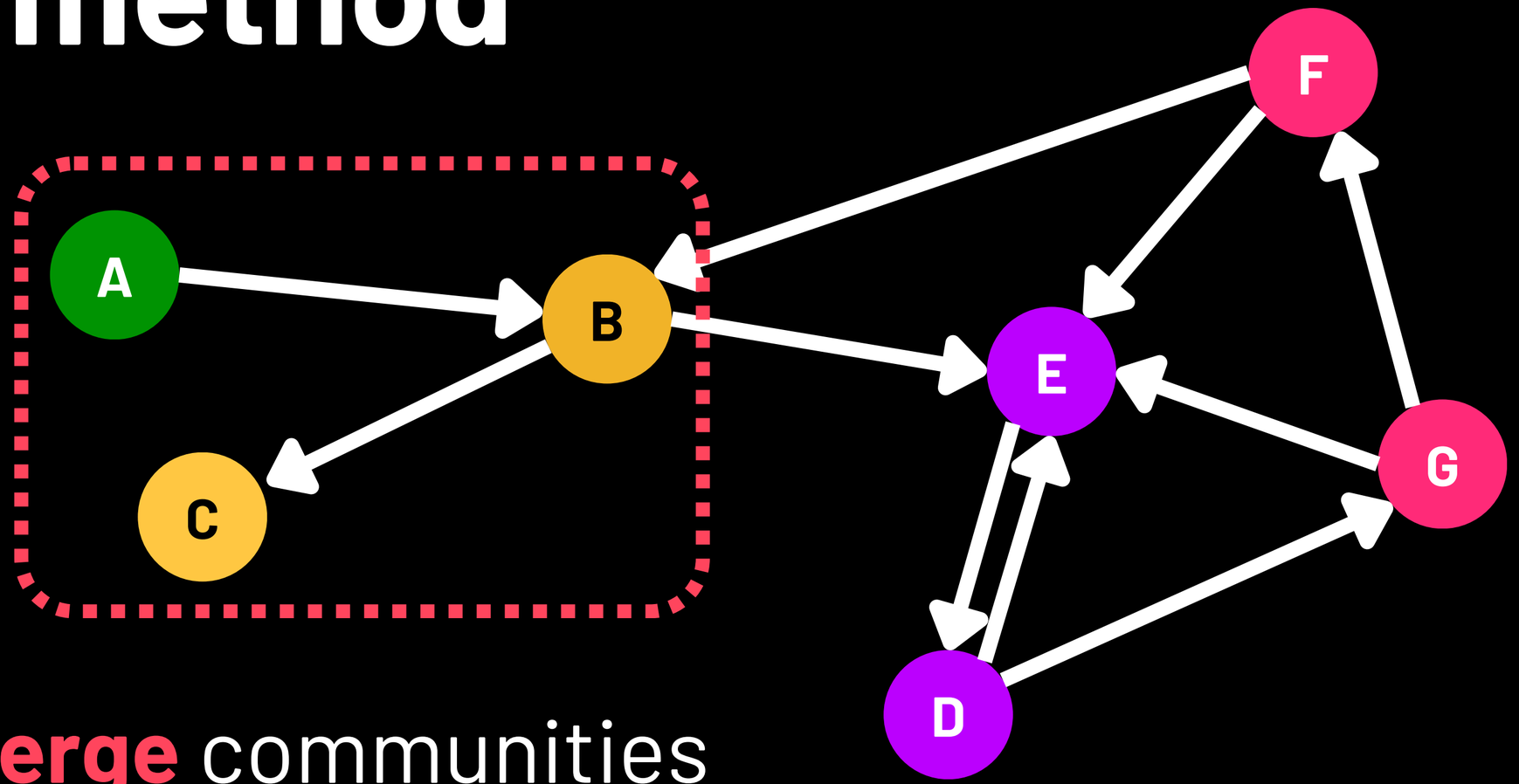


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

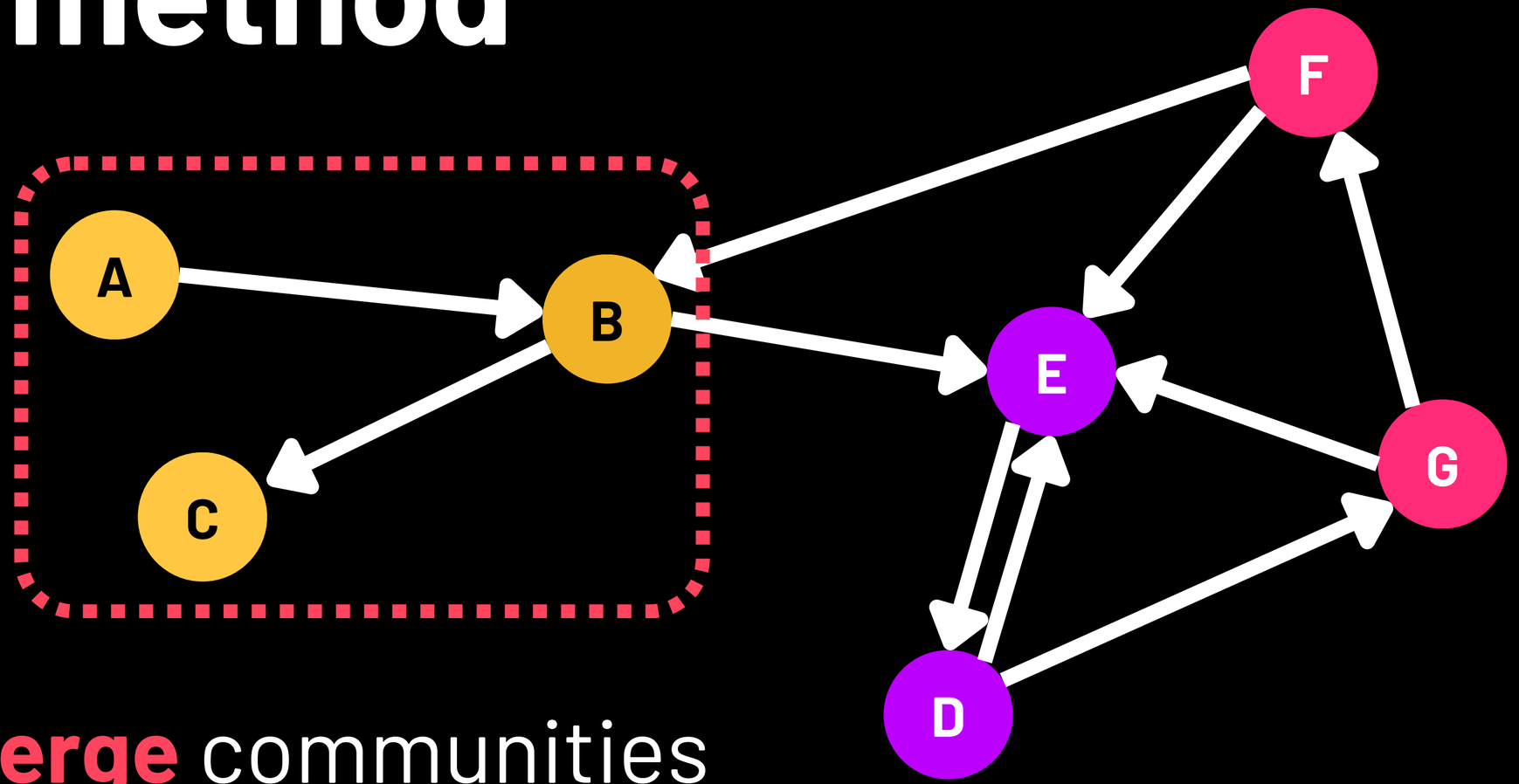


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method

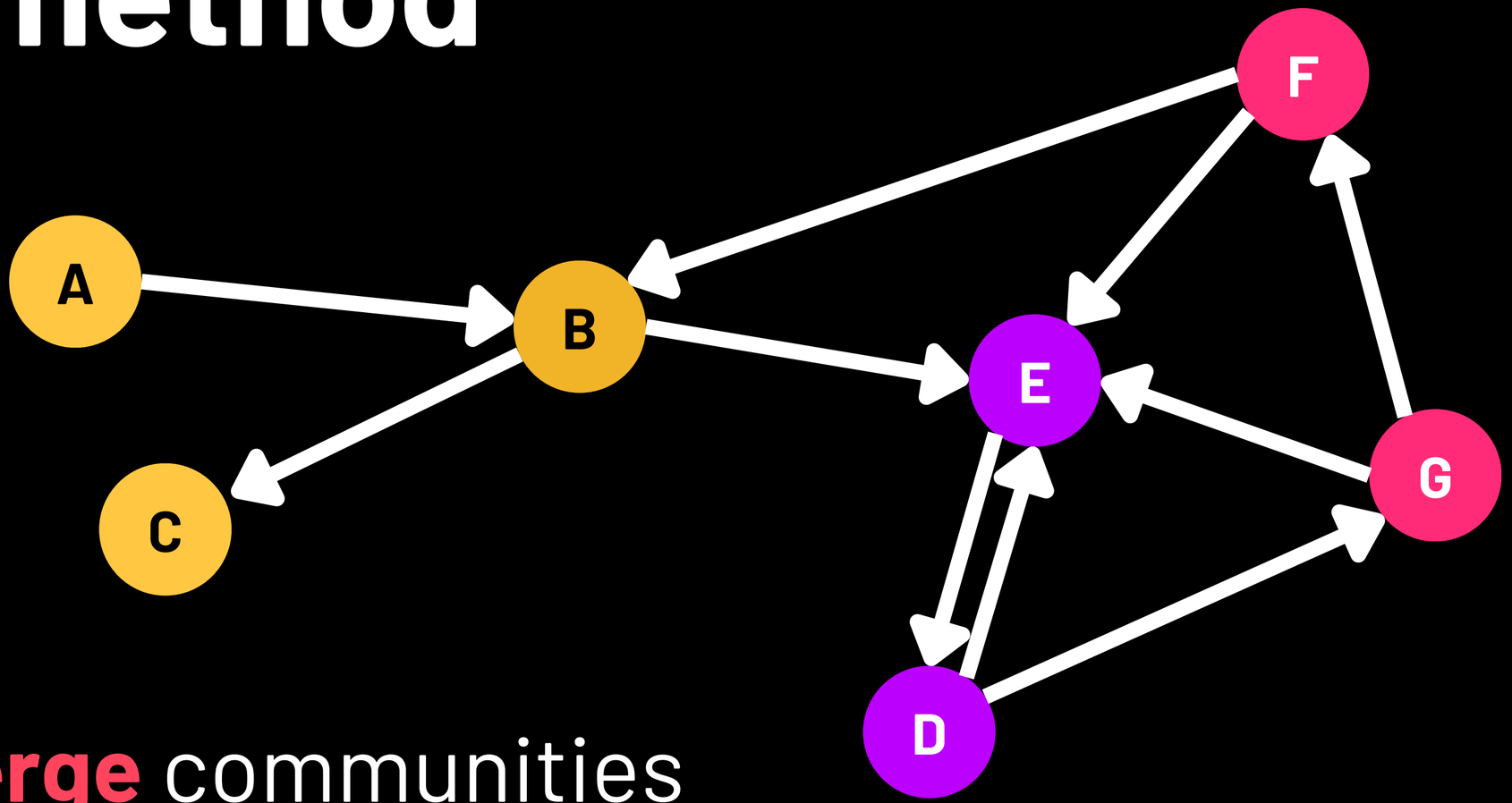


Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

Louvain method



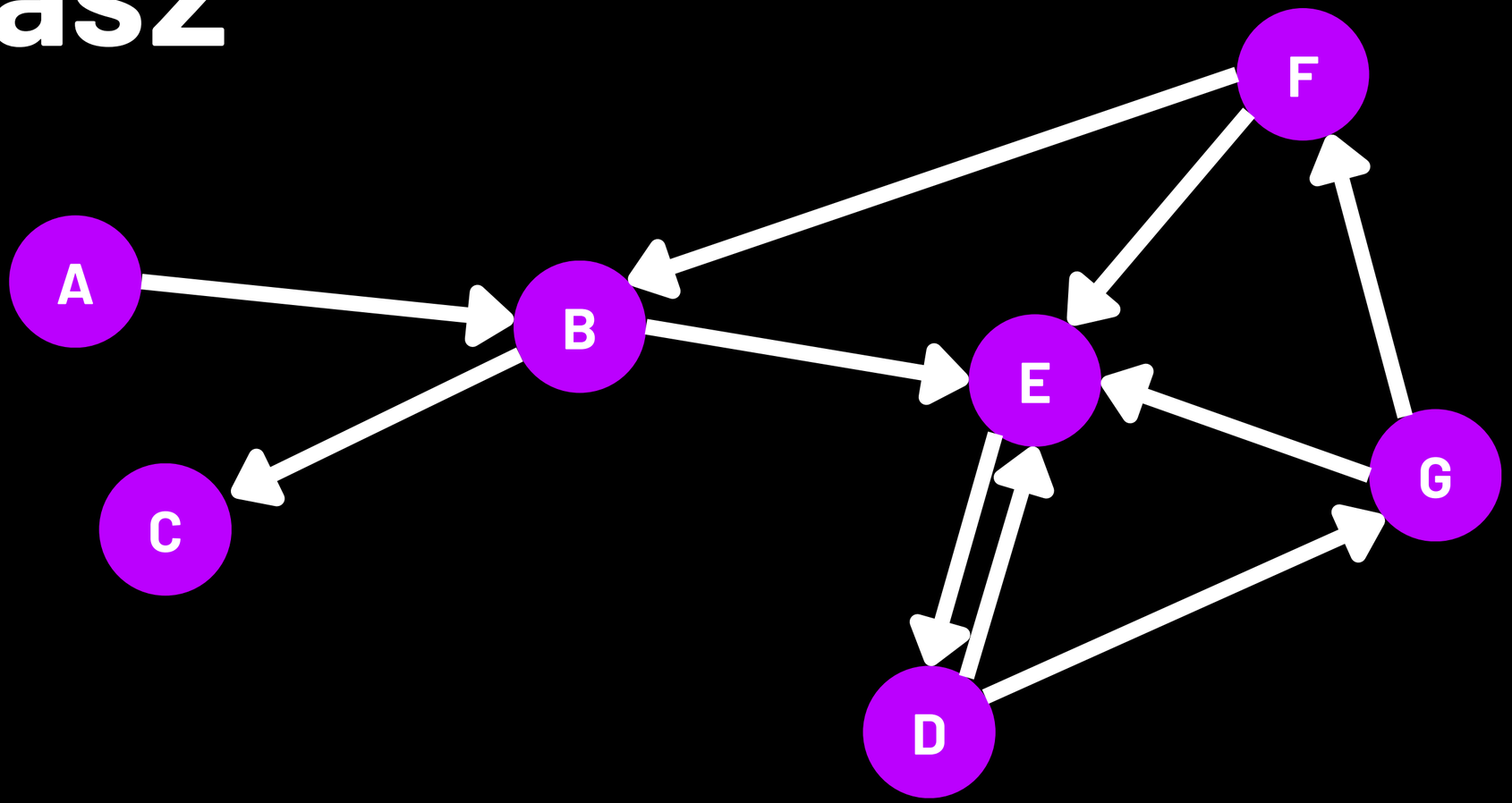
Then, try to **merge** communities
Objective:

- Maximize edges within groups
- Minimize edges between groups

AGORATLAS

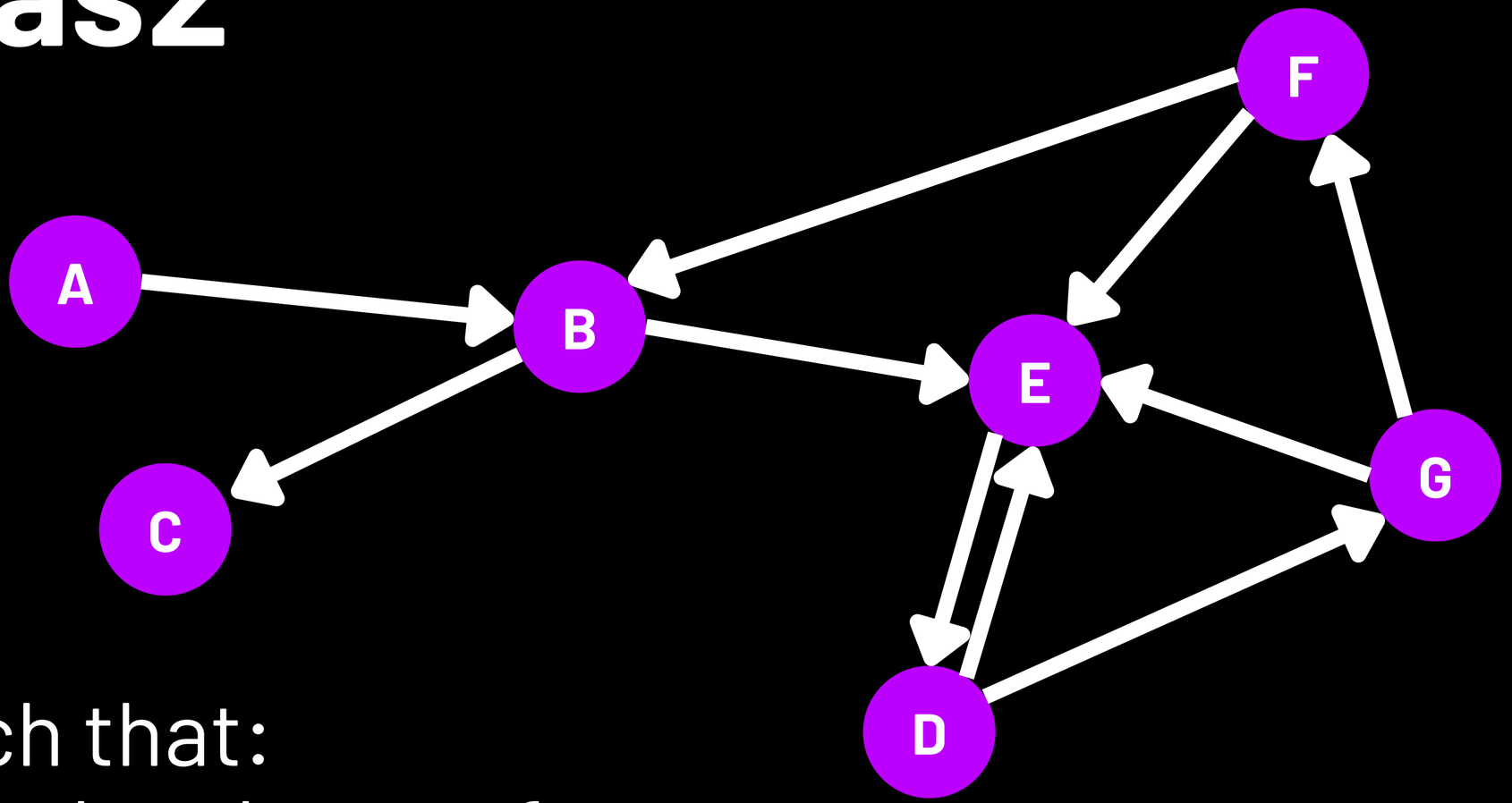
ForceAtlas2

for the 2D layout



AGORATLAS

ForceAtlas2



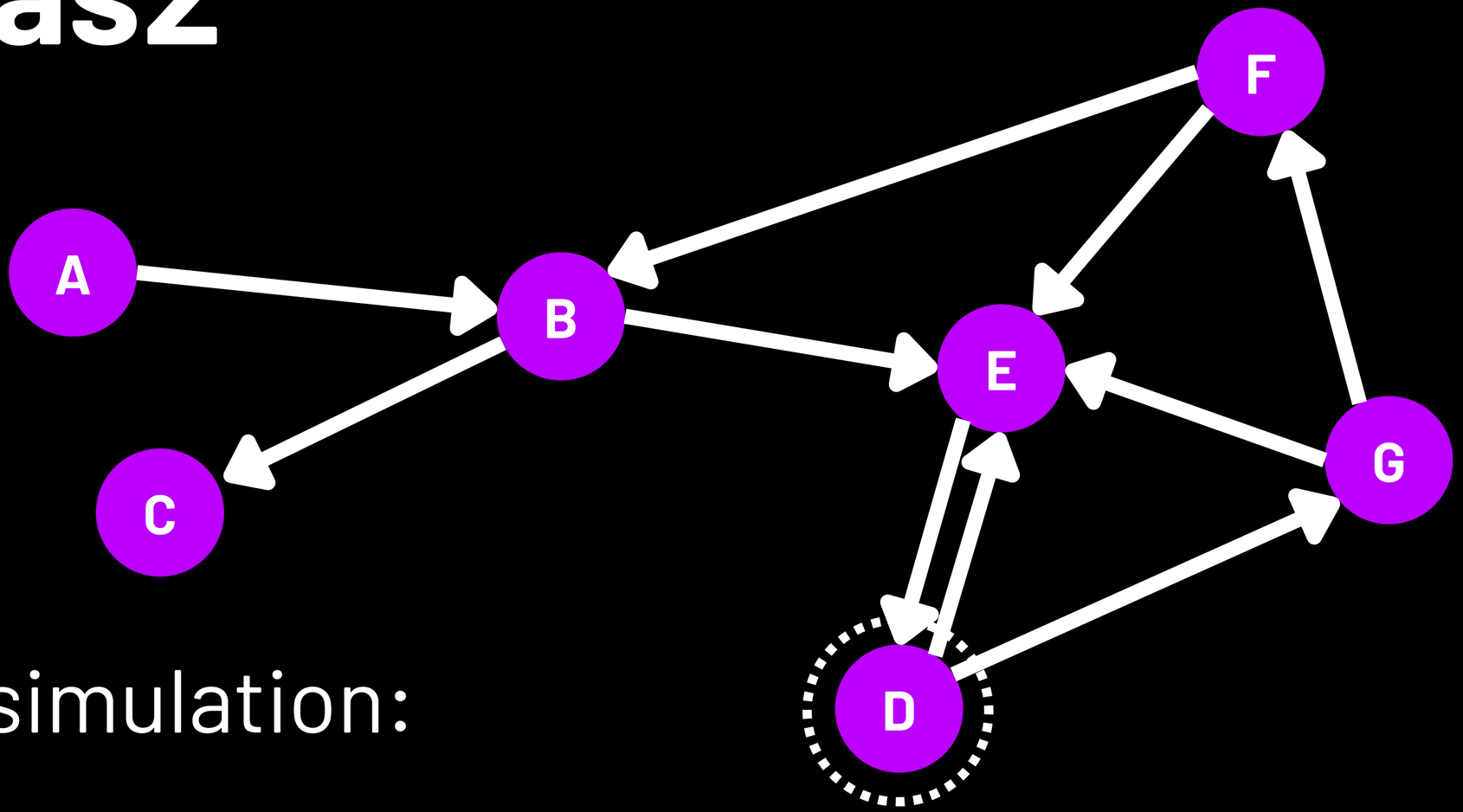
Move nodes such that:

- Disconnected nodes are far apart
- Connected nodes are near each other

→ Use **particle simulation**!

AGORATLAS

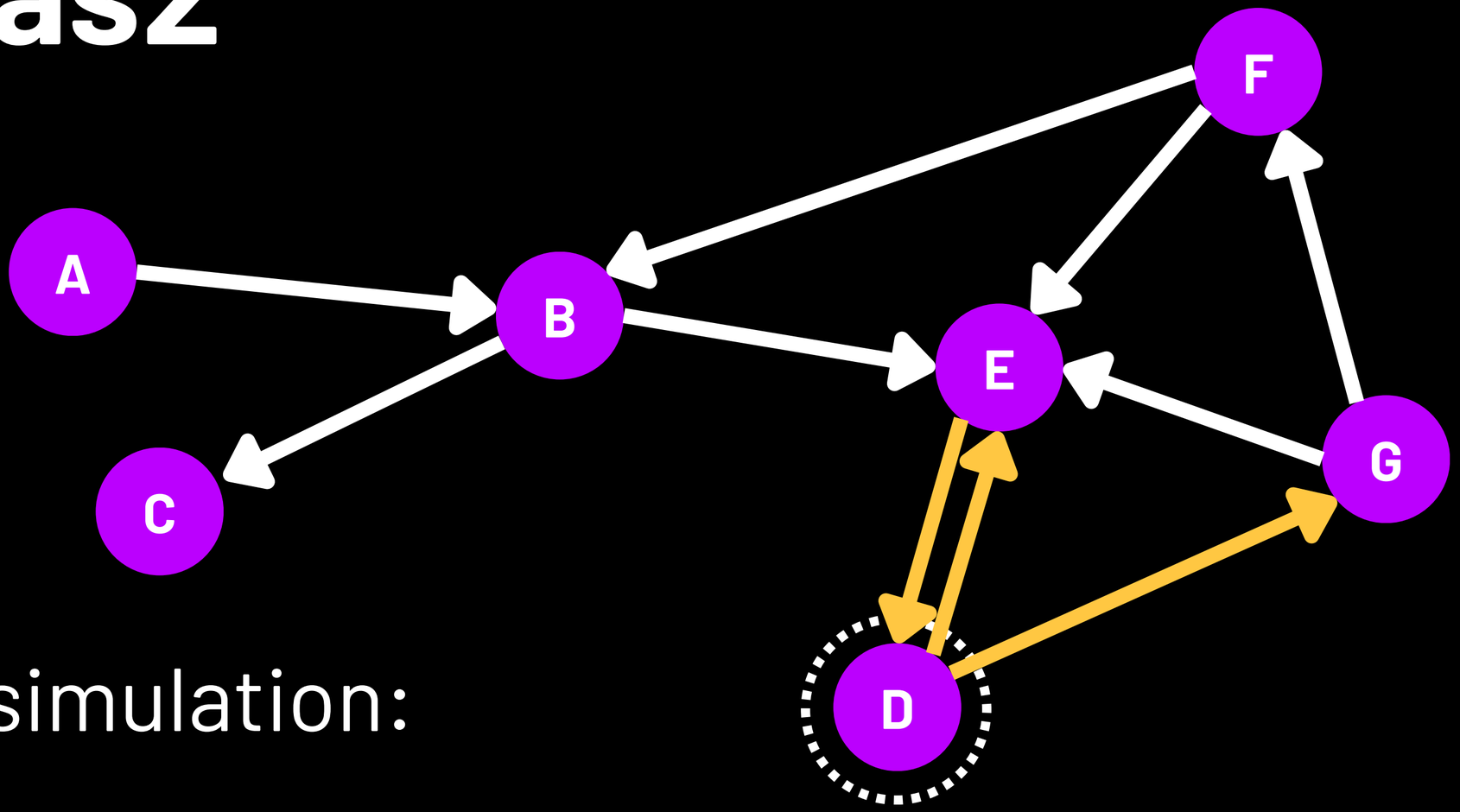
ForceAtlas2



3 forces in the simulation:

AGORATLAS

ForceAtlas2

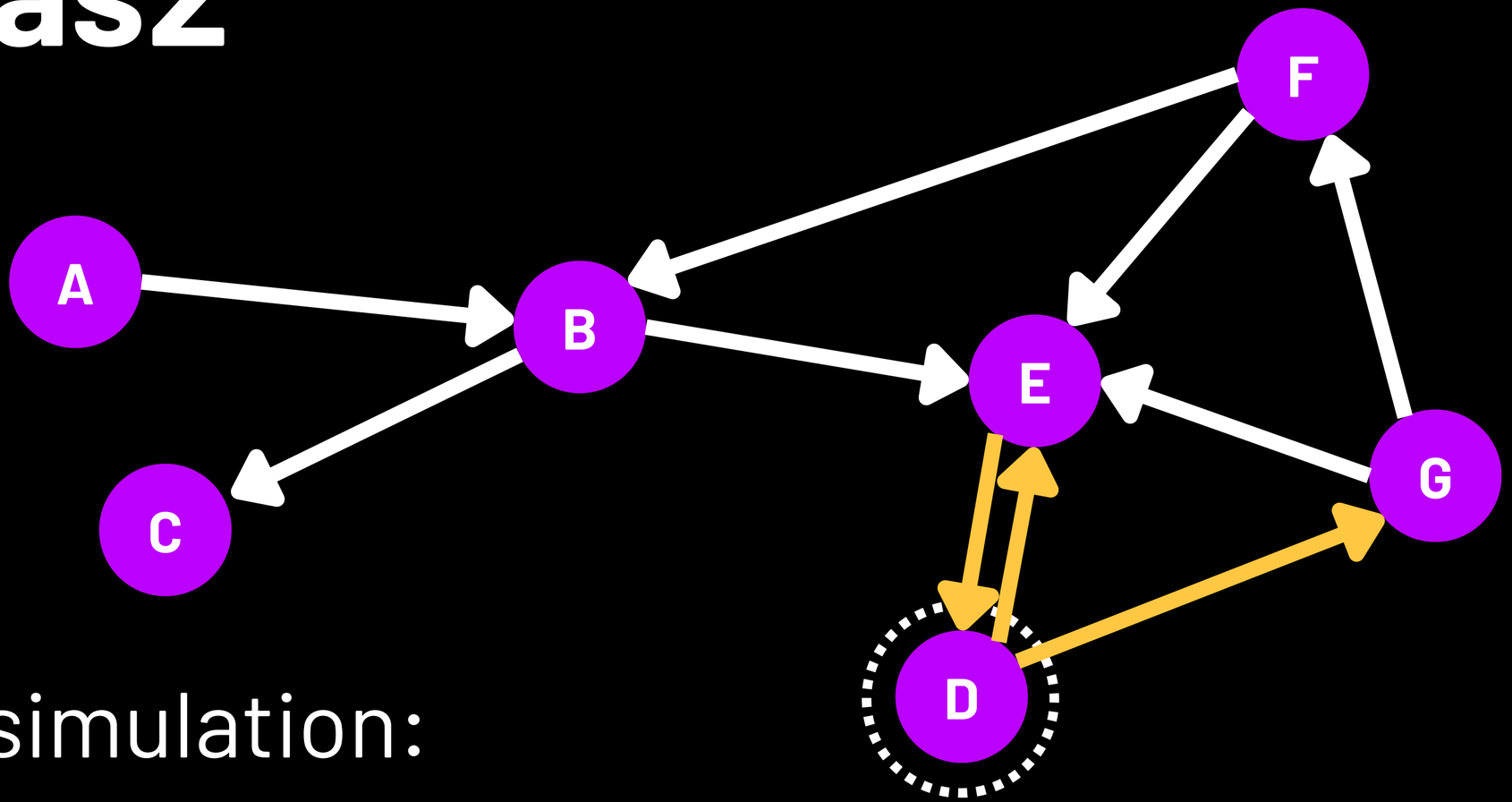


3 forces in the simulation:

- **Attraction**

AGORATLAS

ForceAtlas2

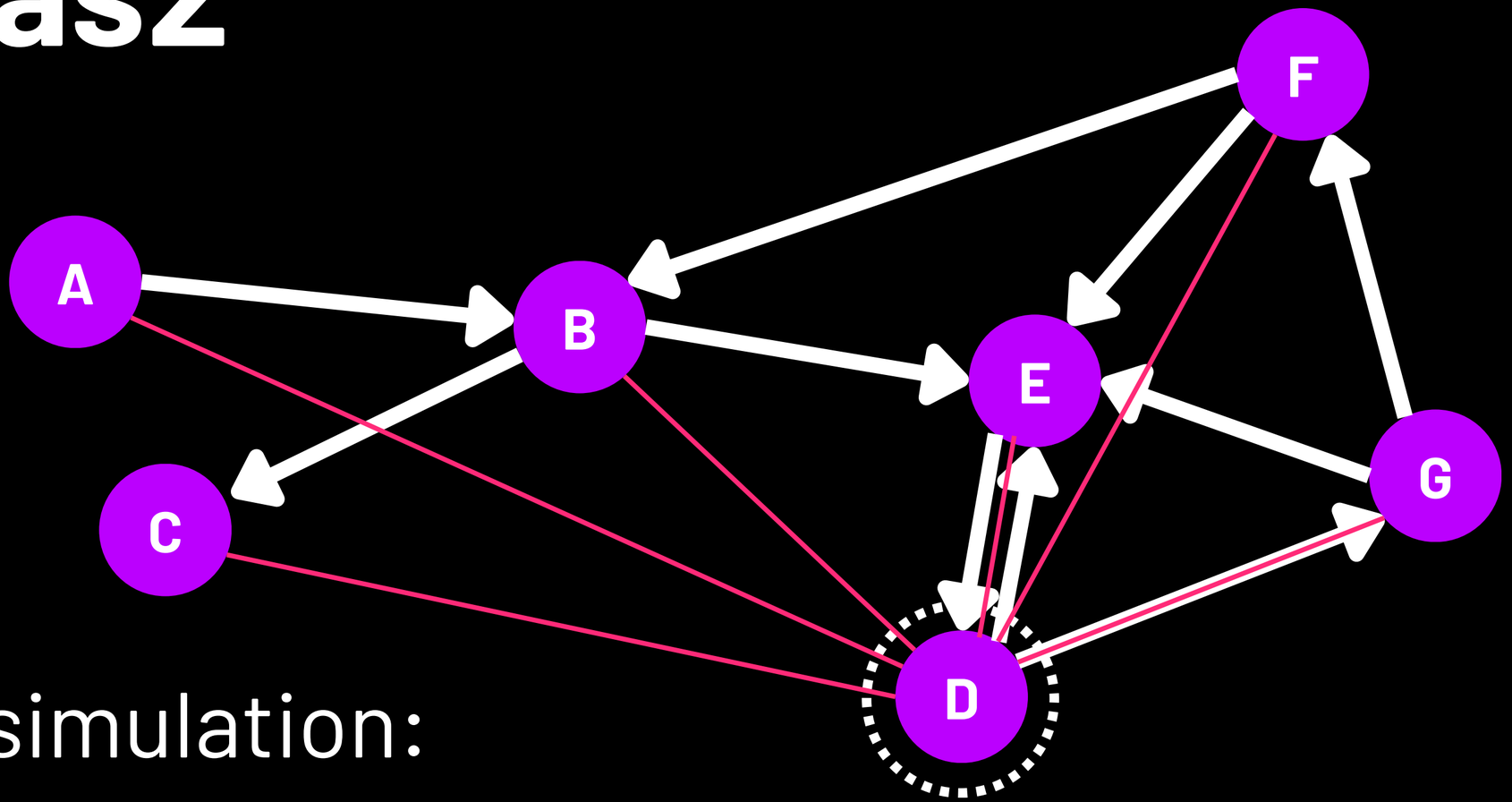


3 forces in the simulation:

- **Attraction**

AGORATLAS

ForceAtlas2

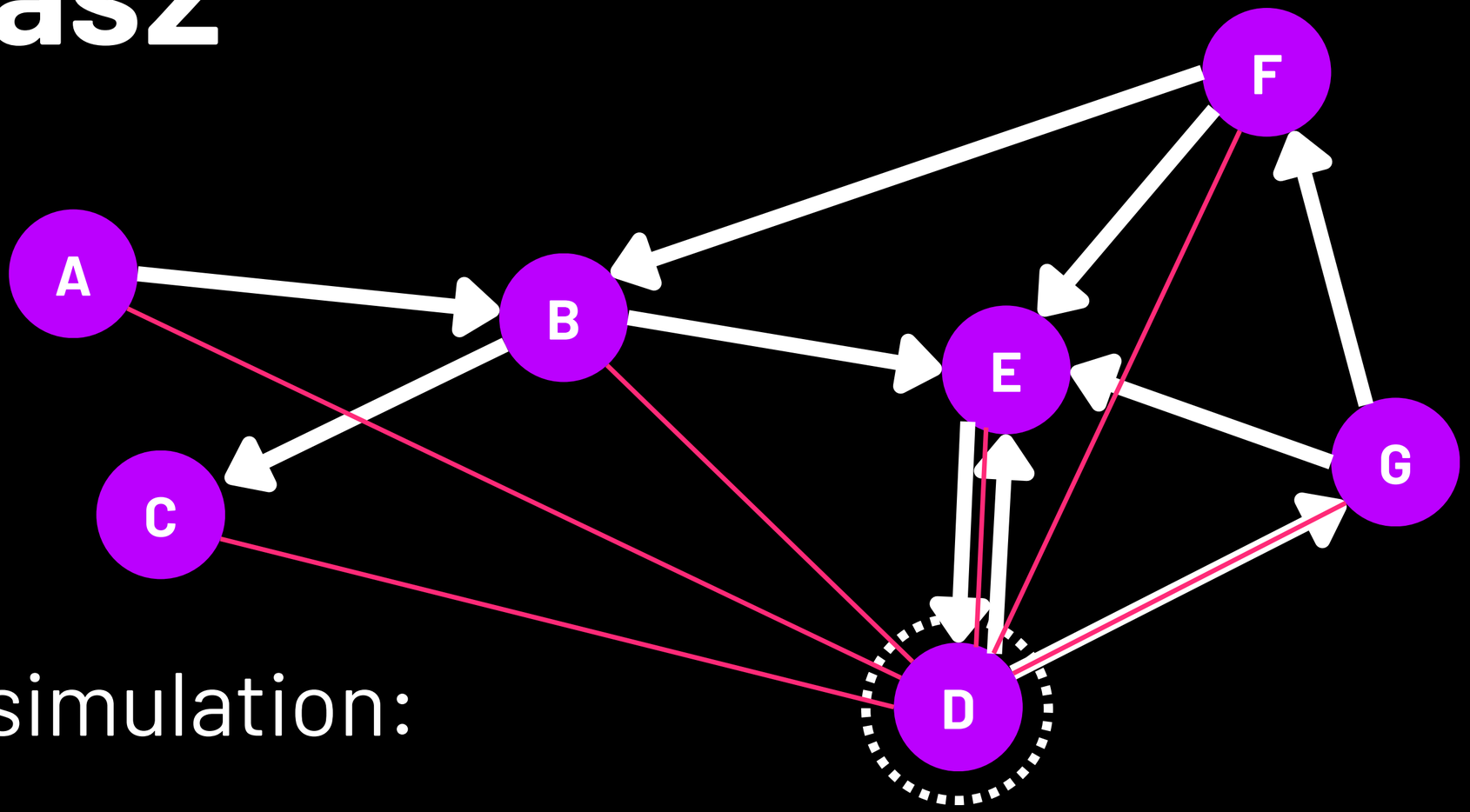


3 forces in the simulation:

- Attraction
- **Repulsion**

AGORATLAS

ForceAtlas2

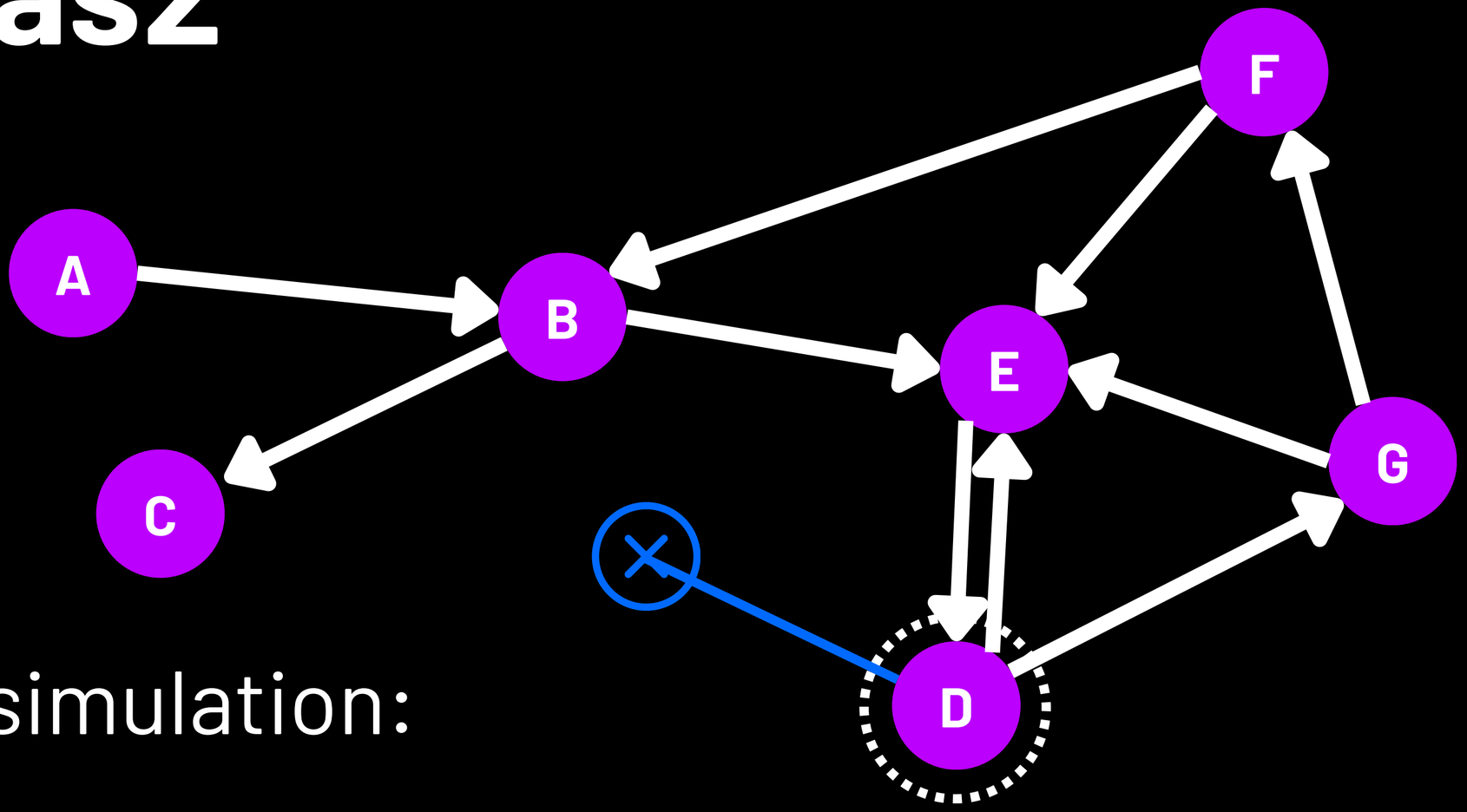


3 forces in the simulation:

- Attraction
- **Repulsion**

AGORATLAS

ForceAtlas2

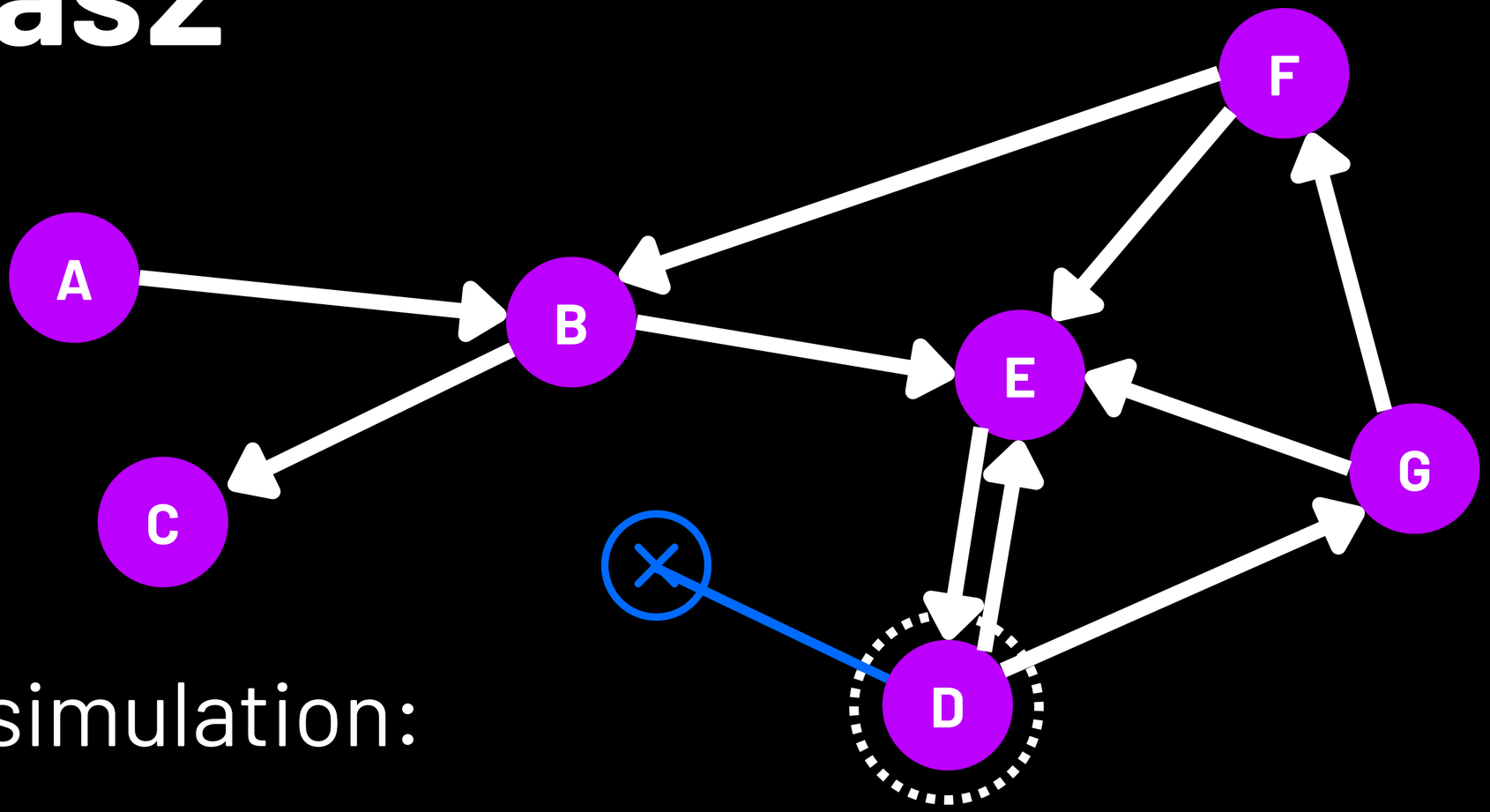


3 forces in the simulation:

- Attraction
- Repulsion
- **Gravity**

AGORATLAS

ForceAtlas2

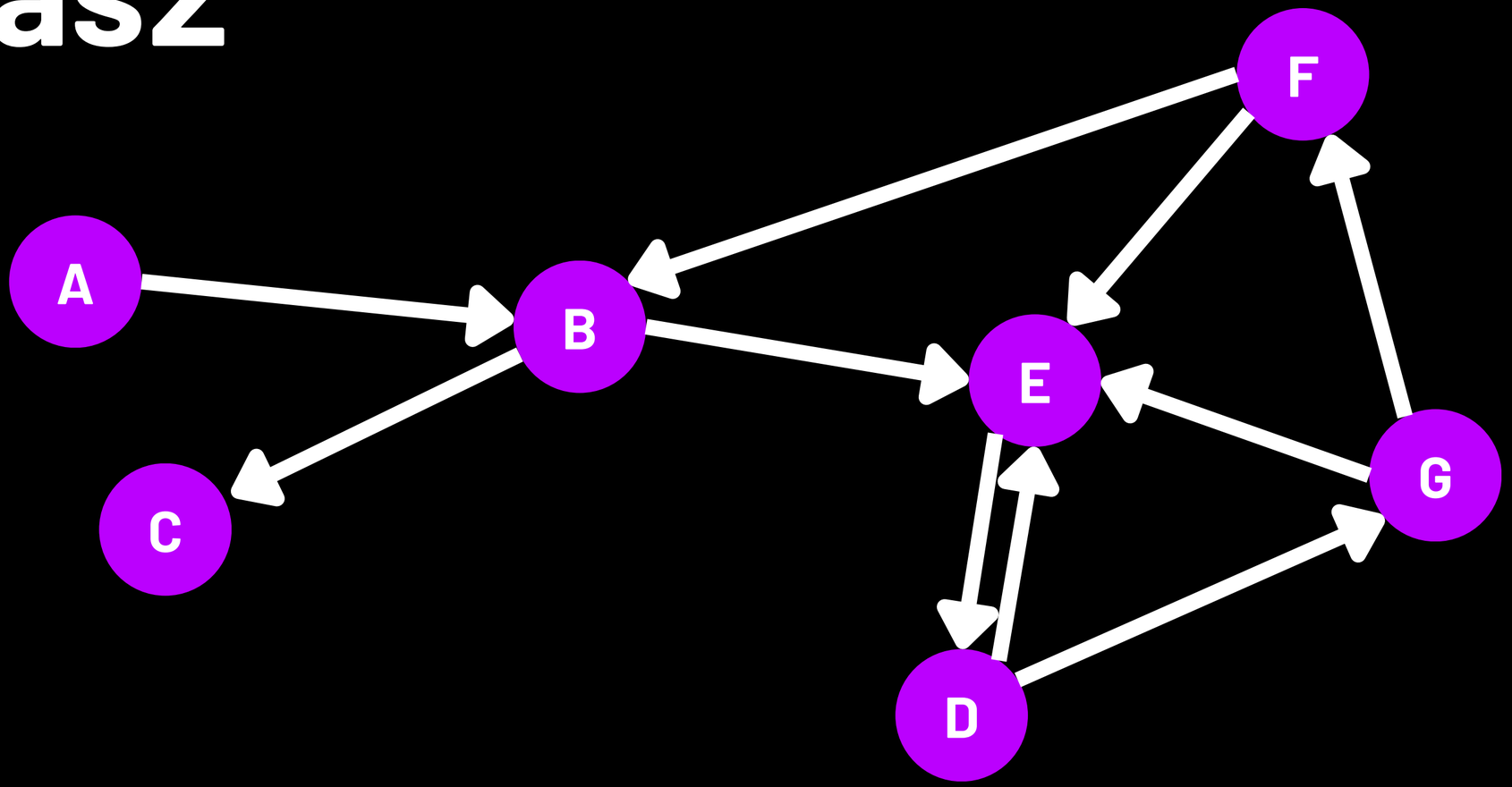


3 forces in the simulation:

- Attraction
- Repulsion
- **Gravity**

AGORATLAS

ForceAtlas2



Repeat on all nodes for **100+ iterations**

AGORATLAS

Why is everything so slow?

Understanding complexity

Why is everything so slow?

Understanding complexity

Algorithms consume inputs and produce outputs

Computation time (aka “complexity”) depends on properties of the input, especially its **size**

Why is everything so slow?

Understanding complexity

Complexity of graph algorithms usually depends on:

- **V**, number of nodes (vertices)
- **E**, number of edges

Why is everything so slow?

Understanding complexity

- **PageRank** has O(**V+E**) complexity

→ A graph 1,000x larger will take
~**1,000x** as much time

Why is everything so slow?

Understanding complexity

- PageRank has O(V+E) complexity
- **Louvain** has O(**E log E**) complexity

→ A graph 1,000x larger will take
~**10,000x** as much time

Why is everything so slow?

Understanding complexity

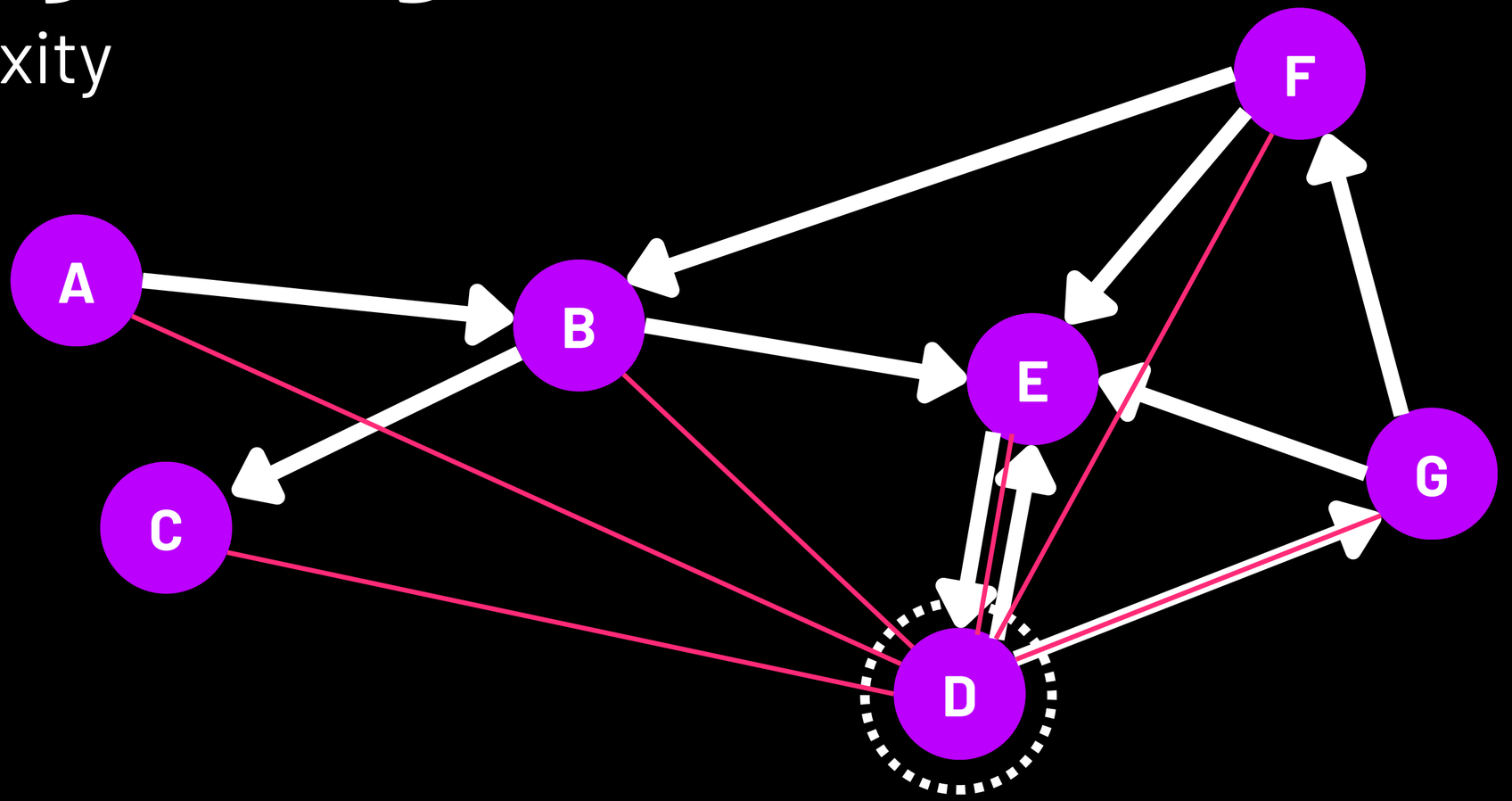
- PageRank has O(V+E) complexity
- Louvain has O(E log E) complexity
- **ForceAtlas2** has O(**V^2+E**) complexity

→ A graph 1,000x larger will take

~**1,000,000x** as much time!

Why is everything so slow?

Understanding complexity



Remember the repulsion force?

Each node interacts with **all** other nodes!

AGORATLAS

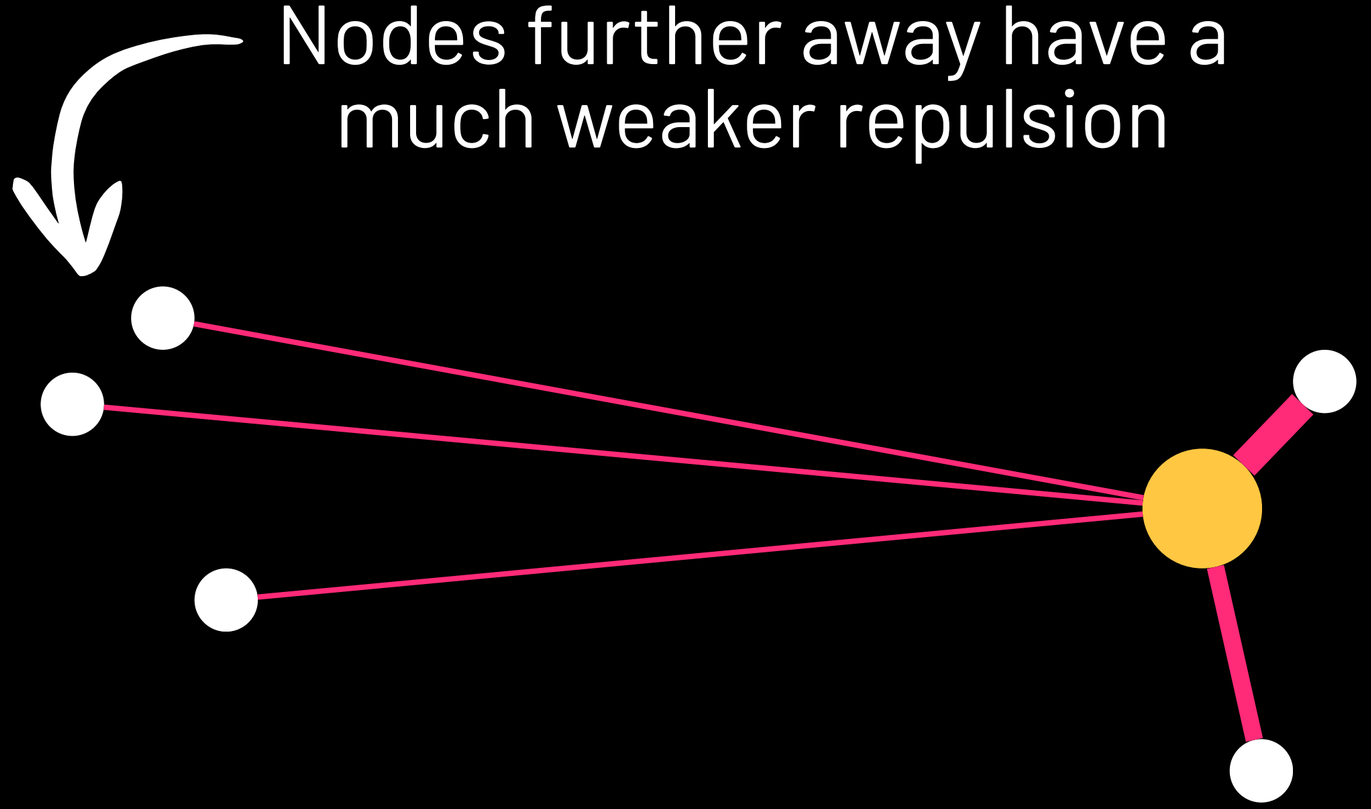
ForceAtlas2 optimization

This problem was solved 40 years ago by **astrophysicists**!

Simulating gravity between multiple celestial bodies is very similar to the ForceAtlas2 repulsion force

ForceAtlas2 optimization

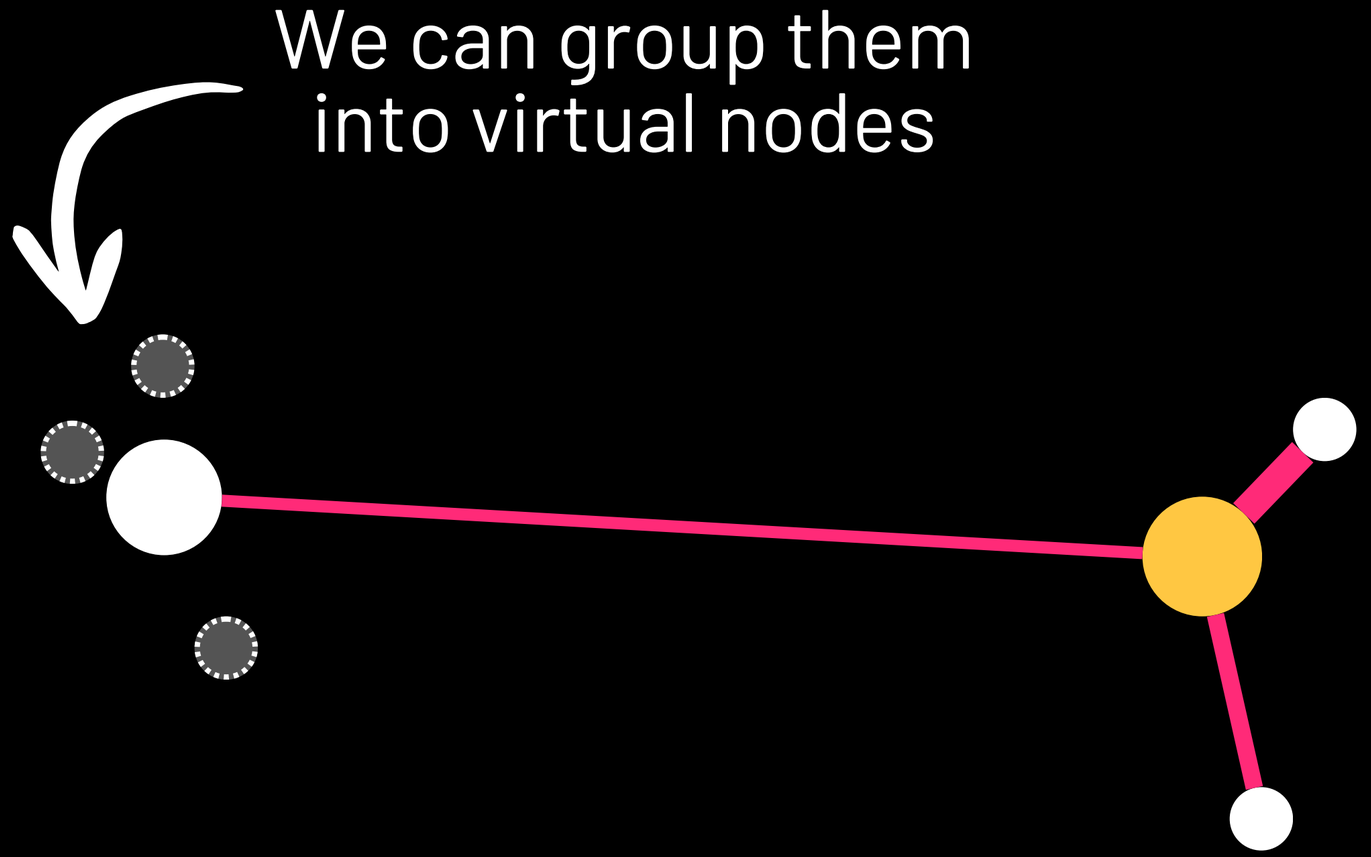
Base observation of the **Barnes-Hut** algorithm:



AGORATLAS

ForceAtlas2 optimization

Base observation of the **Barnes-Hut** algorithm:

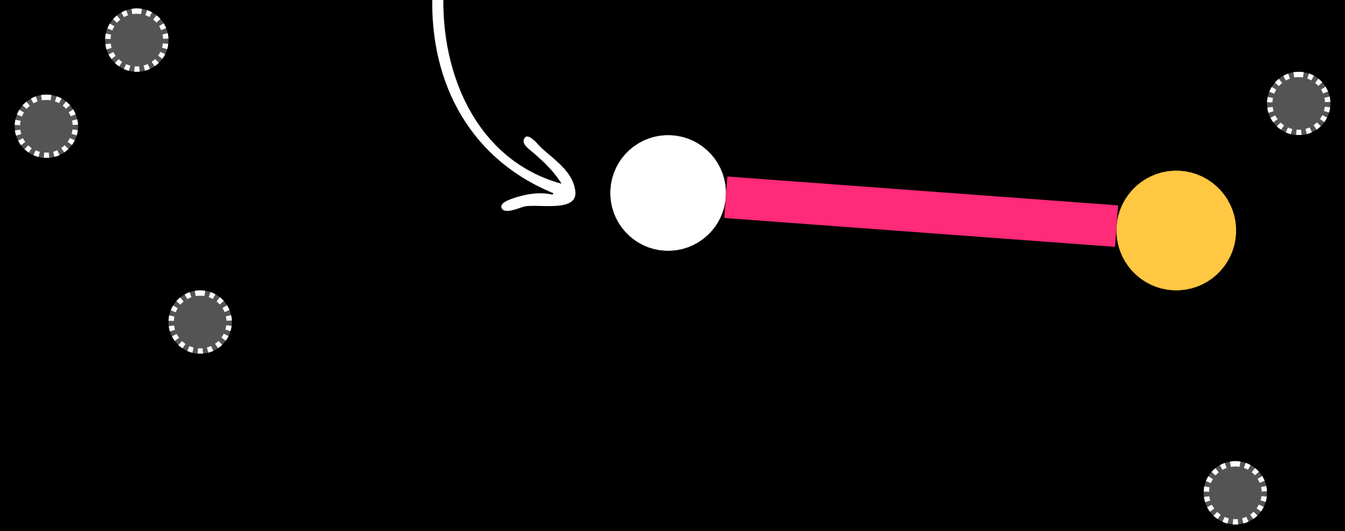


AGORATLAS

ForceAtlas2 optimization

Parameter **theta** determines how large the groups can be

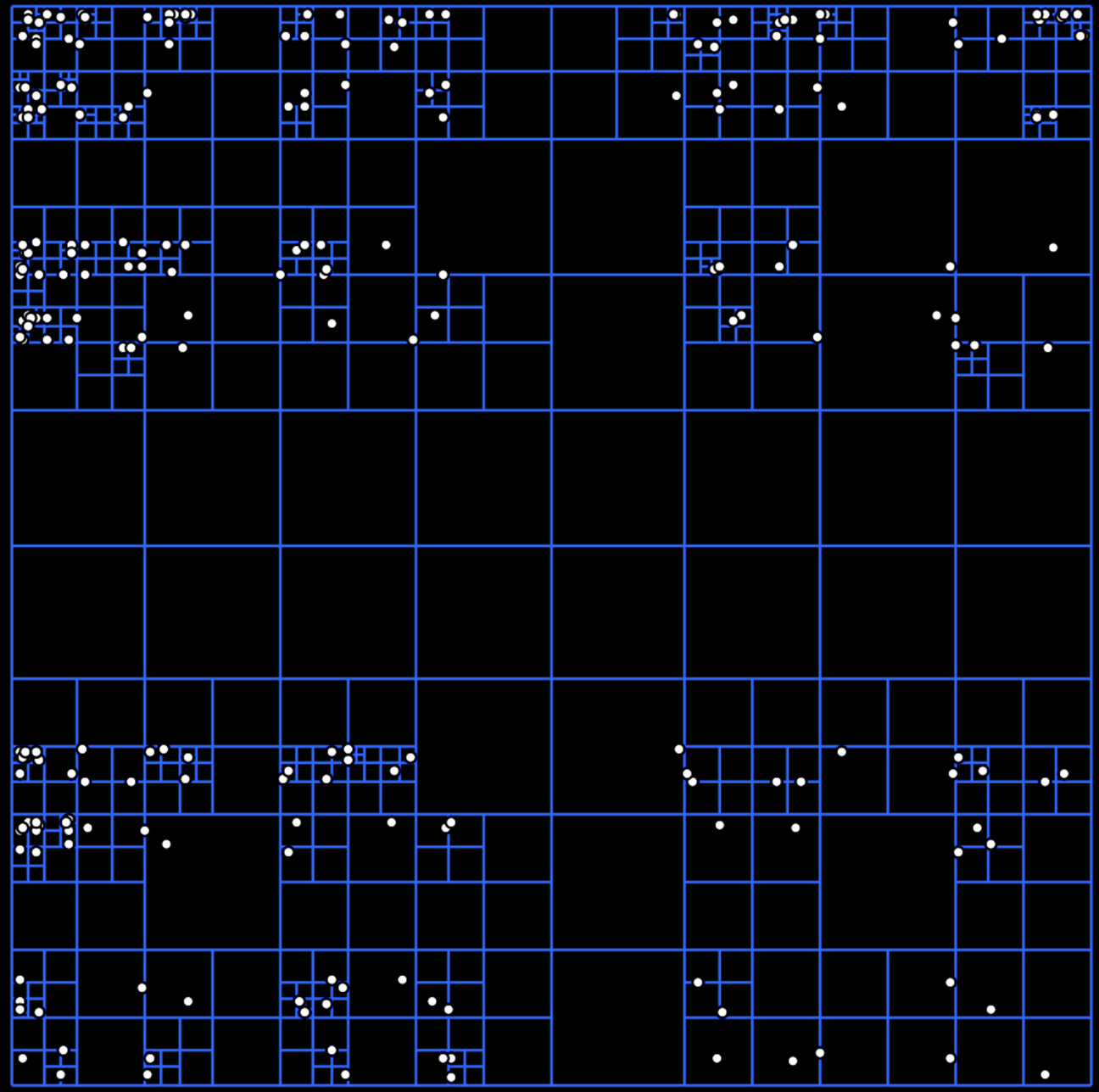
High **theta** = faster but less precise



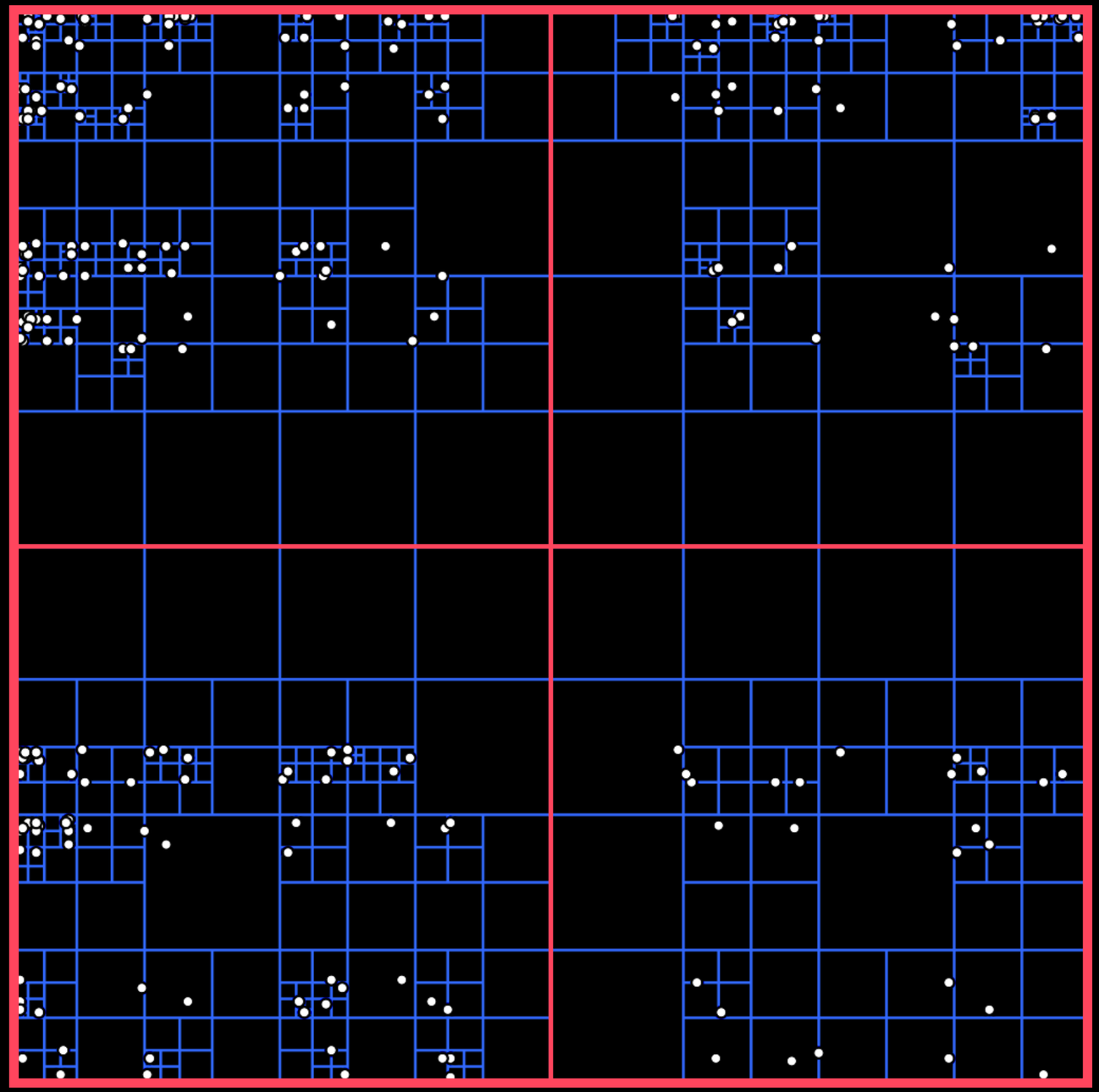
AGORATLAS

The **Quadtree** structure

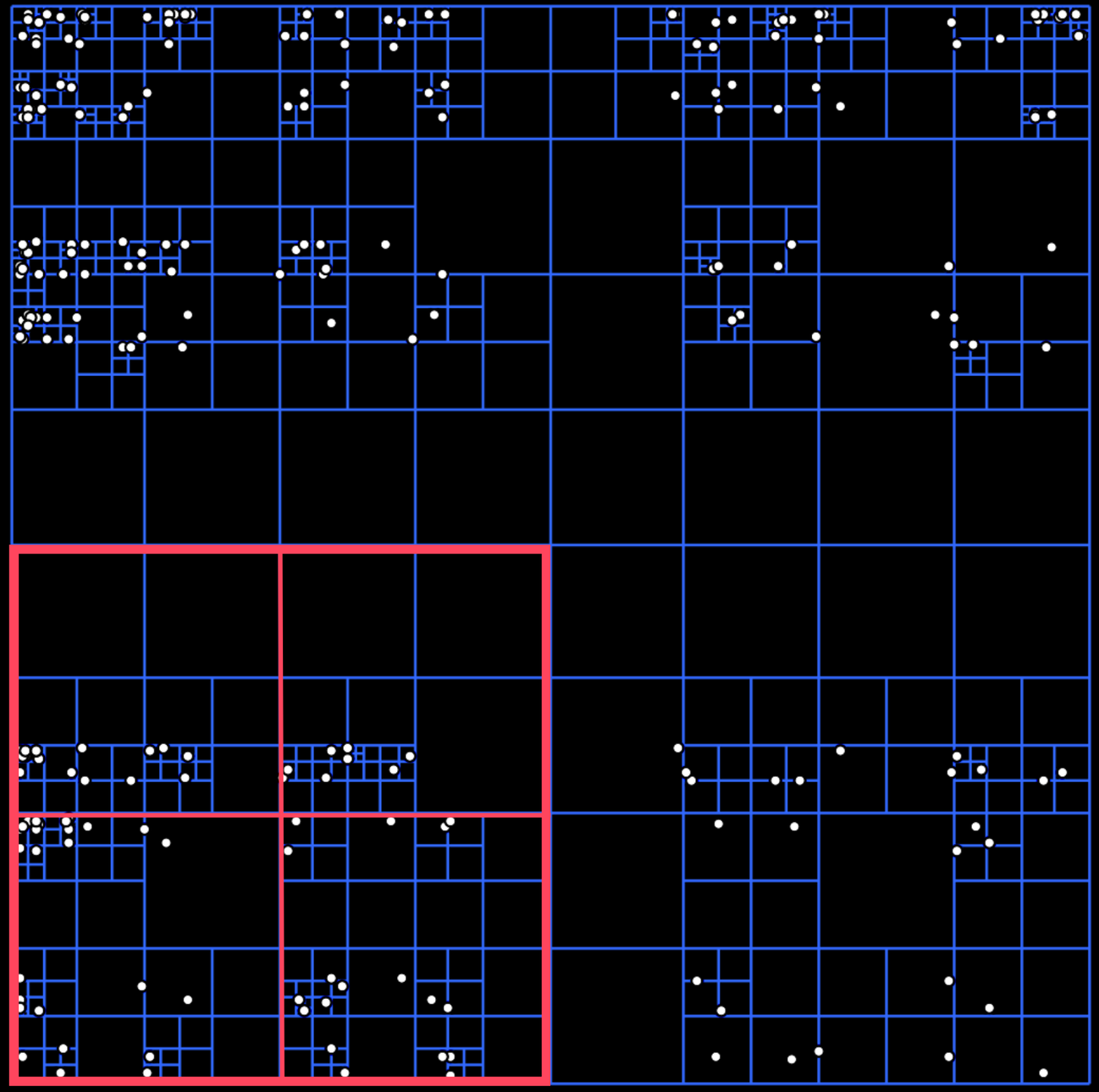
Useful for aggregating data on 2D objects at multiple scales



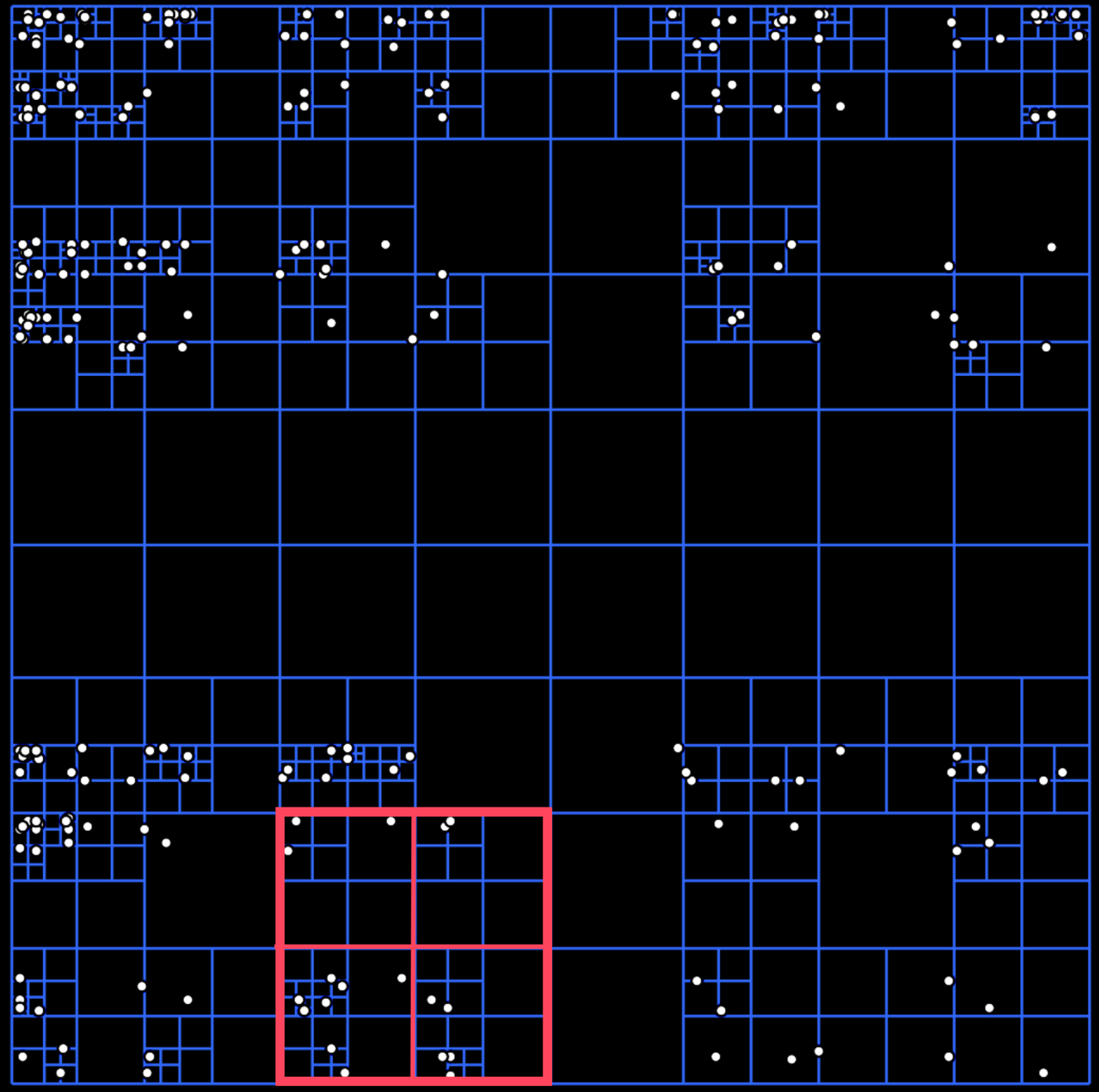
Space is recursively divided in 2x2 cells



Space is recursively divided in 2x2 cells



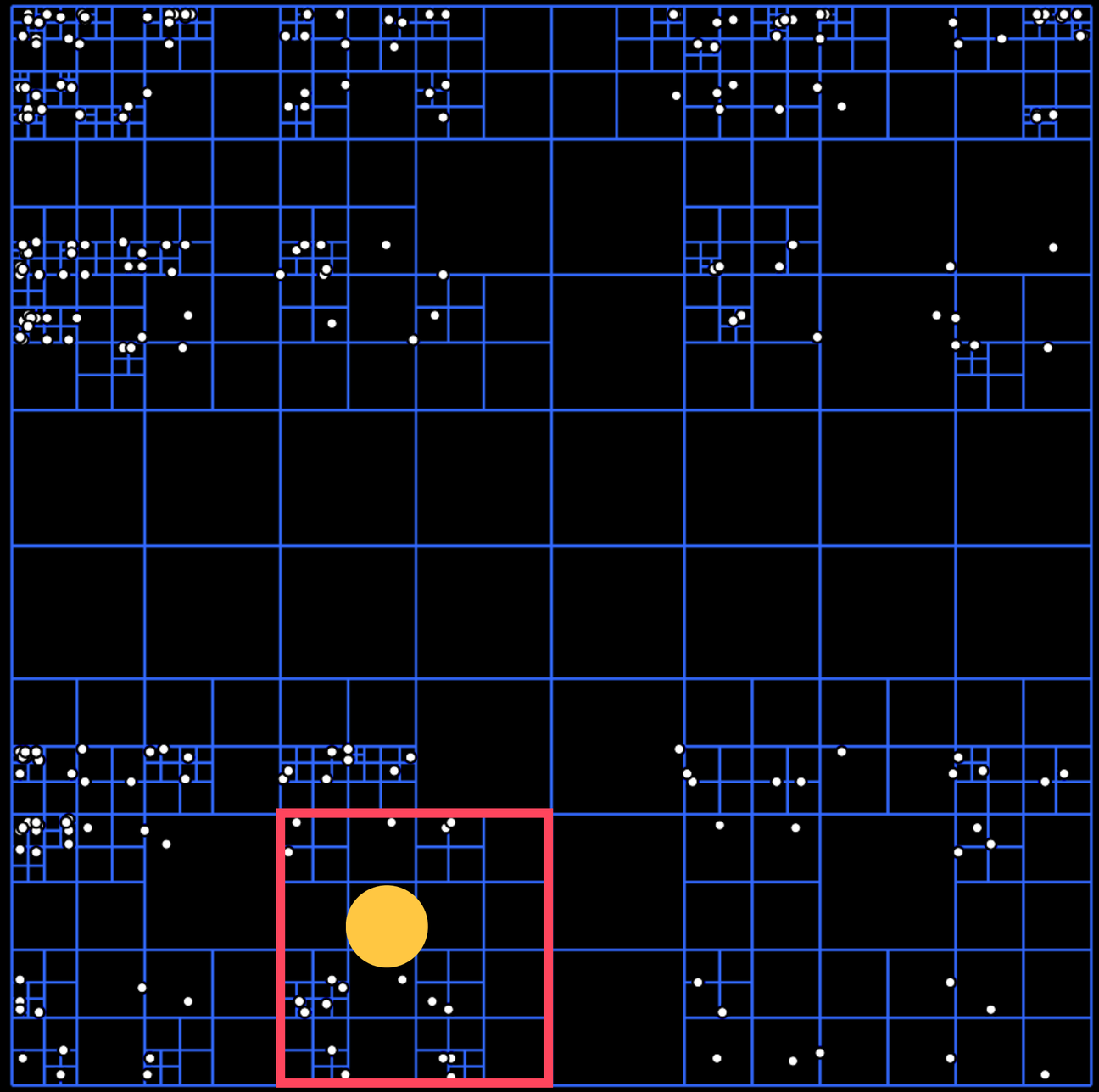
Space is recursively divided in 2x2 cells



Space is recursively divided in 2x2 cells

Each cell contains a **virtual node**

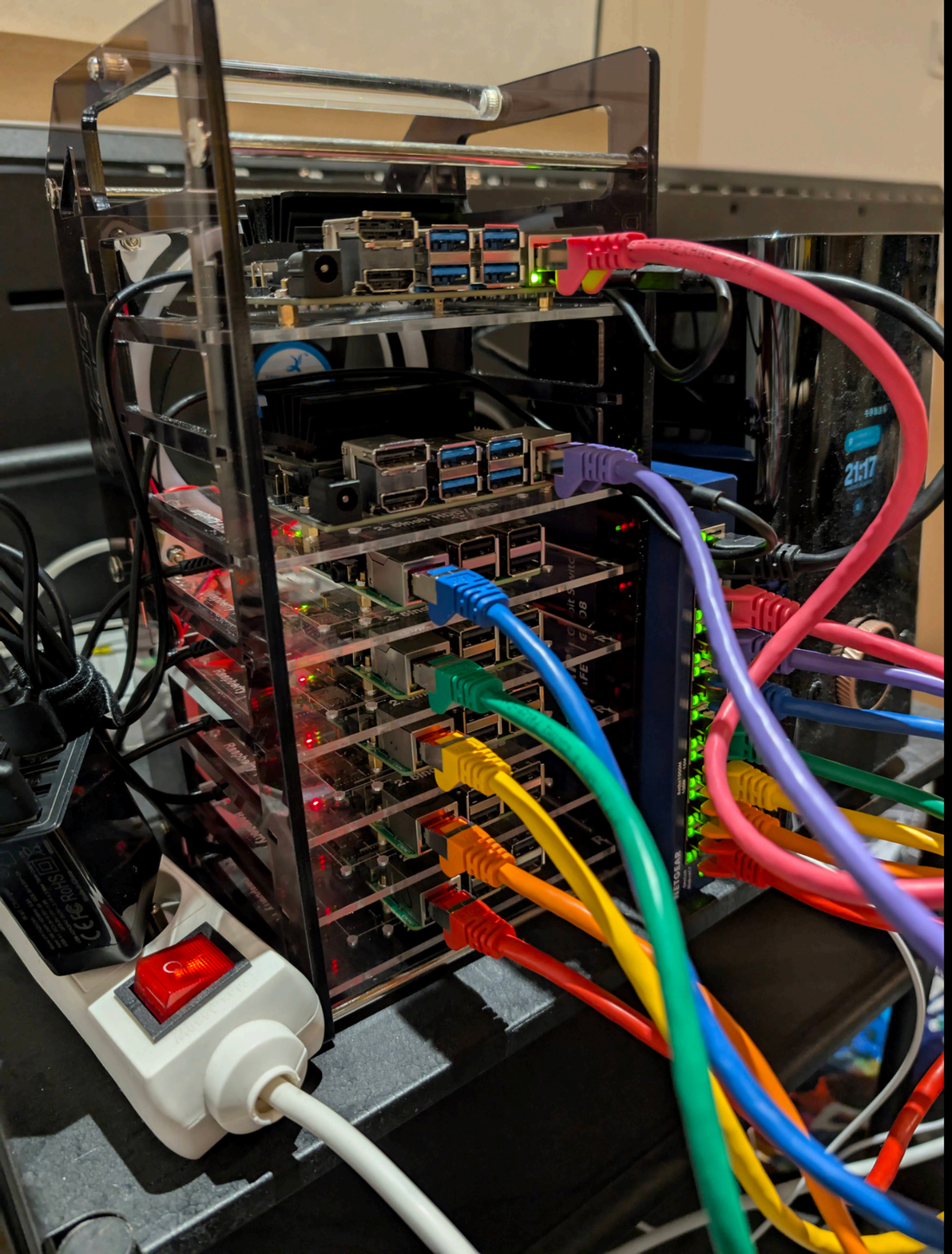
This effectively reduces complexity from $O(V^2)$ to $O(V \log V)$



ForceAtlas2 optimization

- **PageRank** is O(**V+E**)
- **Louvain** is O(**E log E**)
- **ForceAtlas2** is O(**V log V + E**)

ForceAtlas2 is now quasi-linear!



Getting rid of bottlenecks: time for a hardware upgrade!

... yes, our first cluster was literally a bunch of Raspberry Pi stacked together

... and the new one is a bunch of servers mounted in an IKEA table 😇

Reaching the limits of CPU computing

Graphs with 10M nodes are now feasible, but very slow to compute (**>24 hours**)

Adding **GPU** helps a lot, now **100M+** is within reach!

→ We are actively working with NVIDIA on this, the first full implementation of ForceAtlas2 on GPU is now open-source !

But we’re not done yet...

Graph data is now computed, but how do we **visualize** it?

Most languages can natively draw images and shapes

- WebGL
- ImageJ
- Pillow
- Pretty much any UI framework

... but they all break well before **1 million** components 😭

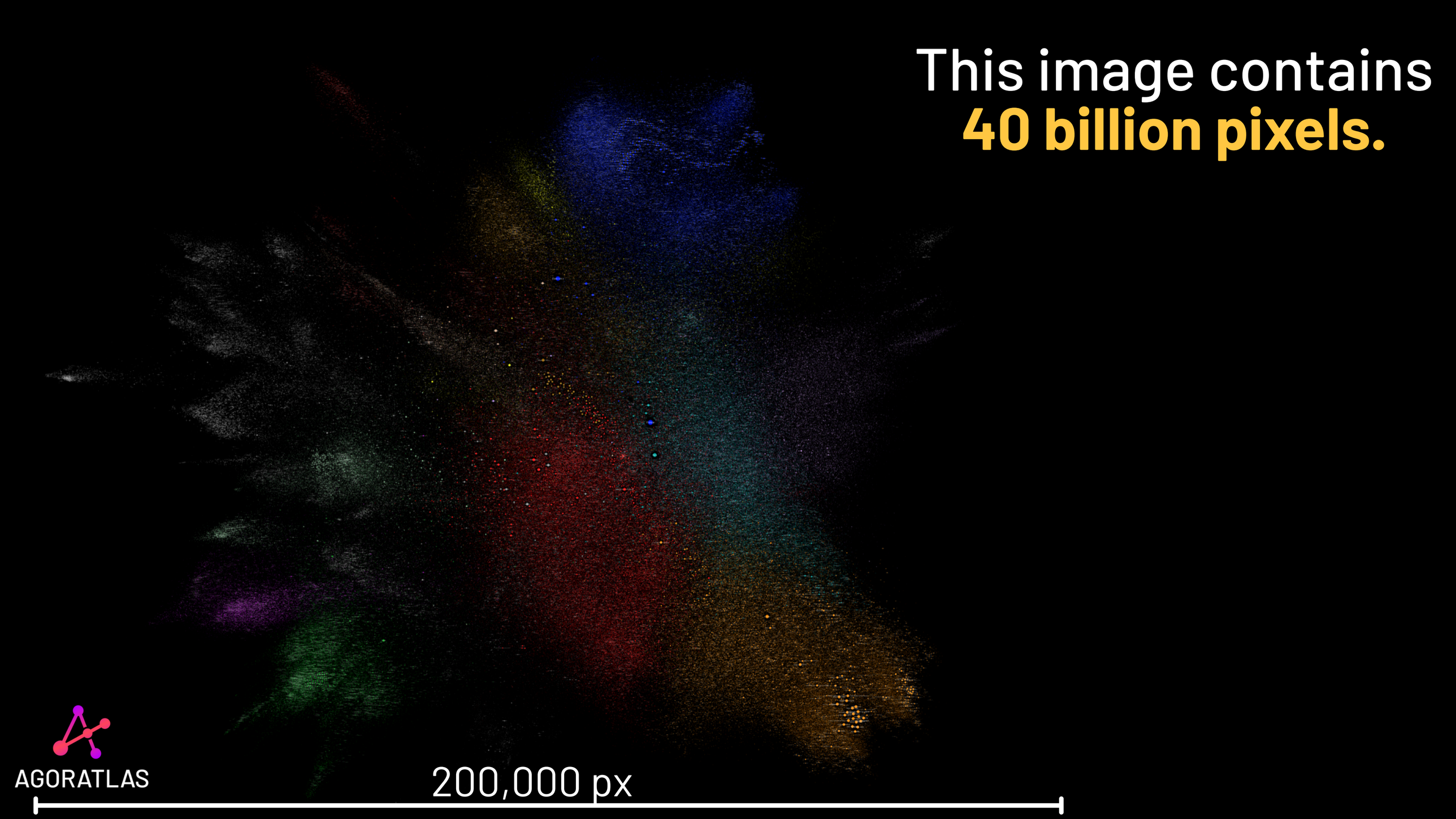
The most viable option is to use **SVG files**

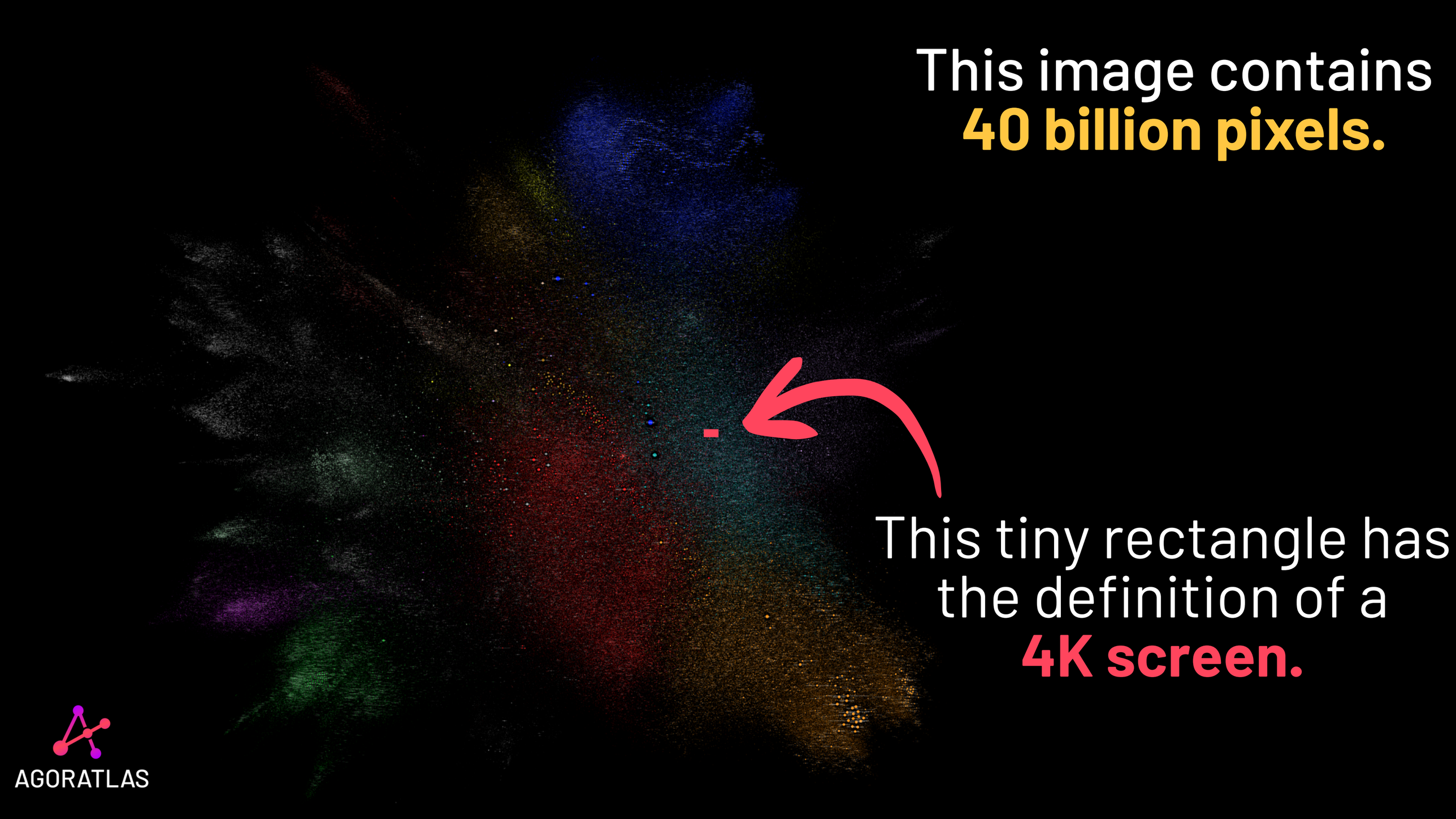
- Vector image format
- Can represent shapes and text
- Essentially an XML file with a list of objects

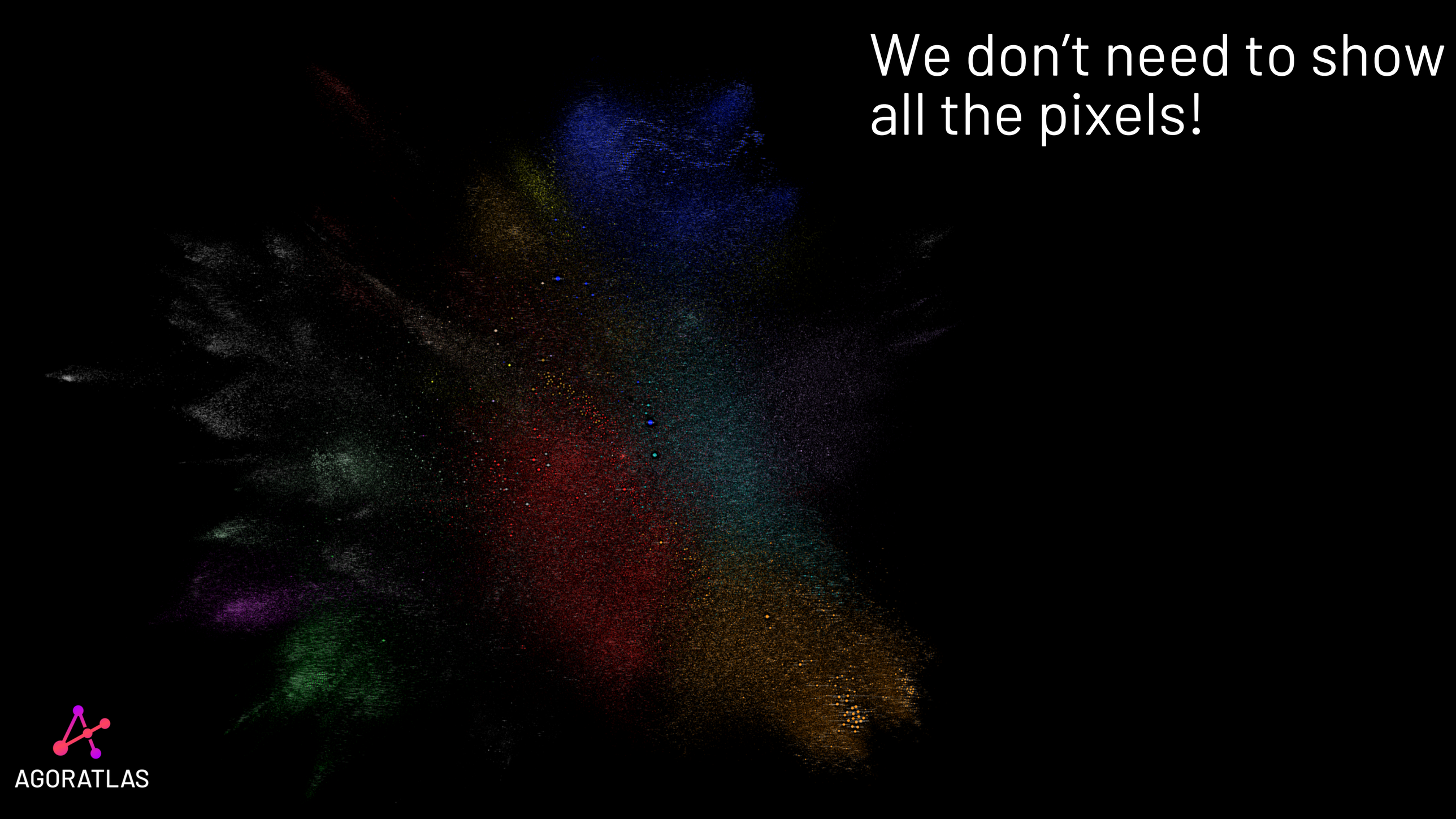
```
<path stroke="#ff0000" stroke-width="0.37106368139083196" opacity="0.5" d="M 48.01771545410156 -157.89361572265625 A 201.86771441186872
<path stroke="#ff0000" stroke-width="0.7765287894989963" opacity="0.5" d="M -80.24983978271484 55.05907440185547 A 39.552963606502054 3
<path stroke="#ff0000" stroke-width="0.7041162119305192" opacity="0.5" d="M -80.24983978271484 55.05907440185547 A 64.50943180528778 64
<path stroke="#ff0000" stroke-width="0.6976015309093256" opacity="0.5" d="M -80.24983978271484 55.05907440185547 A 95.10071654491104 95
<path stroke="#ff0000" stroke-width="0.8376476045550418" opacity="0.5" d="M 46.37086486816406 96.21395111083984 A 42.874585851546875 42
<path stroke="#ff0000" stroke-width="0.7024208173452746" opacity="0.5" d="M 46.37086486816406 96.21395111083984 A 73.41632253994904 73.
<path stroke="#ff0000" stroke-width="0.7765287894989963" opacity="0.5" d="M -11.184531211853027 113.25235748291016 A 32.60020594750644
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="-23.650076" cy="-38.032322" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="-79.122765" cy="8.340253" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="39.565147" cy="4.604787" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="14.810745" cy="-35.875122" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="48.790703" cy="-7.2857323" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="45.615536" cy="37.54264" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="-57.17462" cy="36.555817" r="14.4" fill="#ff0000"/>
<circle stroke="#b20000" stroke-width="3.2" opacity="1.0" cx="-38.098022" cy="31.869427" r="14.4" fill="#ff0000"/>
```

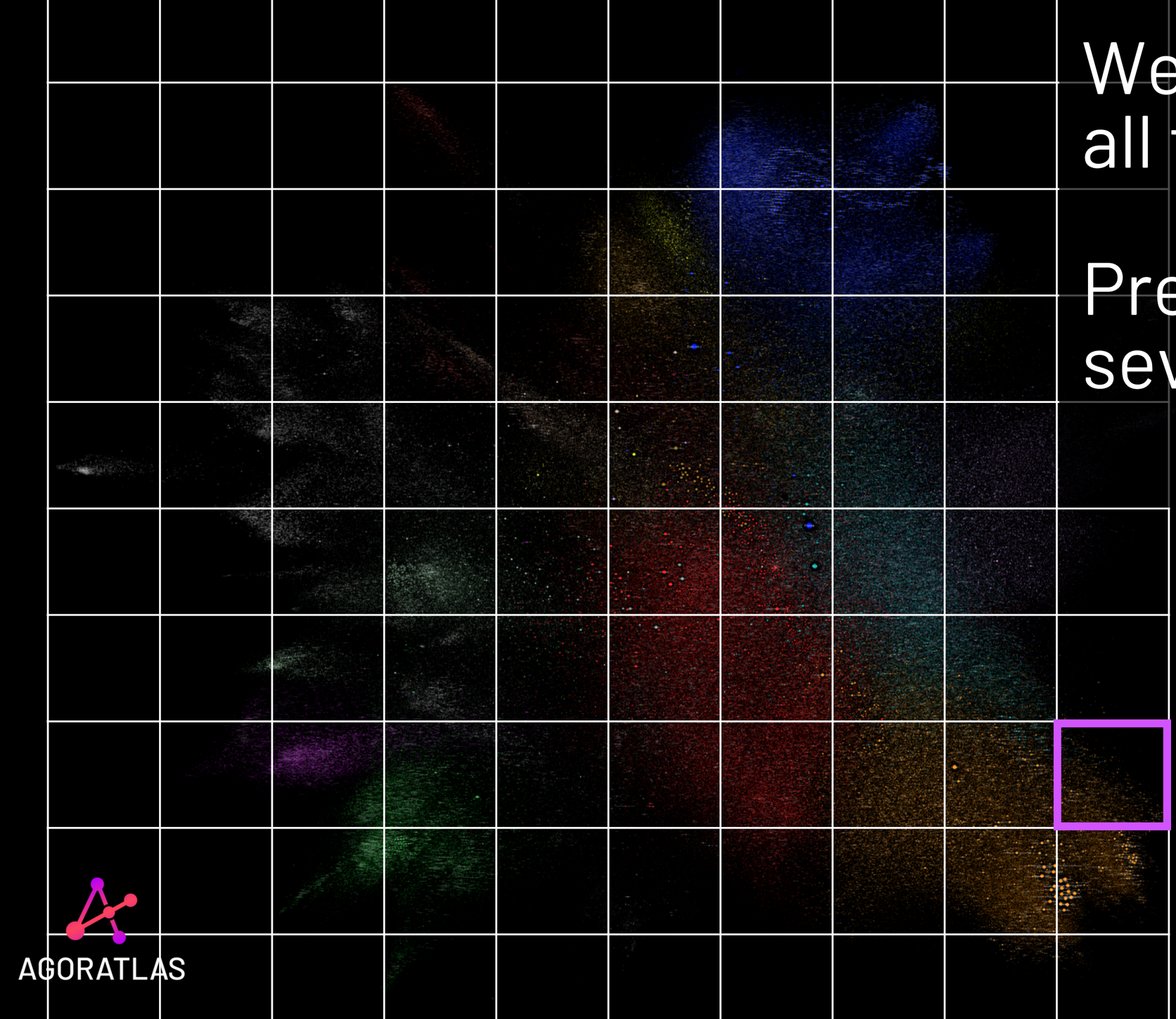
With large graphs, the SVG file can be **several gigabytes** for a single image!

This is still way too much for any normal computer, but we can use a magic trick: **tile pyramids**


This image contains
40 billion pixels.
AGORATLAS
200,000 px


This image contains
40 billion pixels.
This tiny rectangle has
the definition of a
4K screen.
AGORATLAS


We don’t need to show
all the pixels!
AGORATLAS

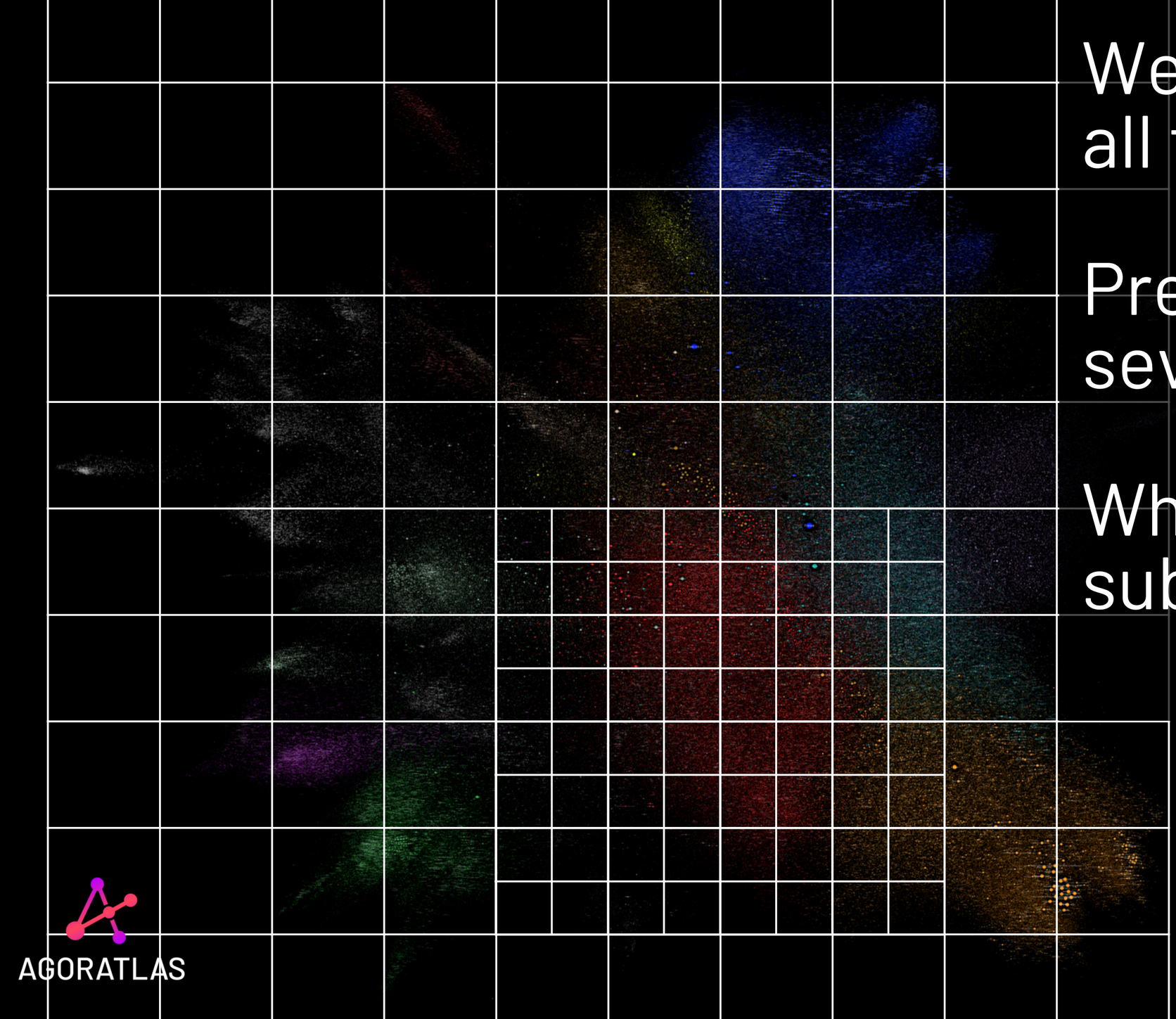


We don’t need to show all the pixels!

Pre-compute **tiles** on several zoom levels

256x256 px

AGORATLAS

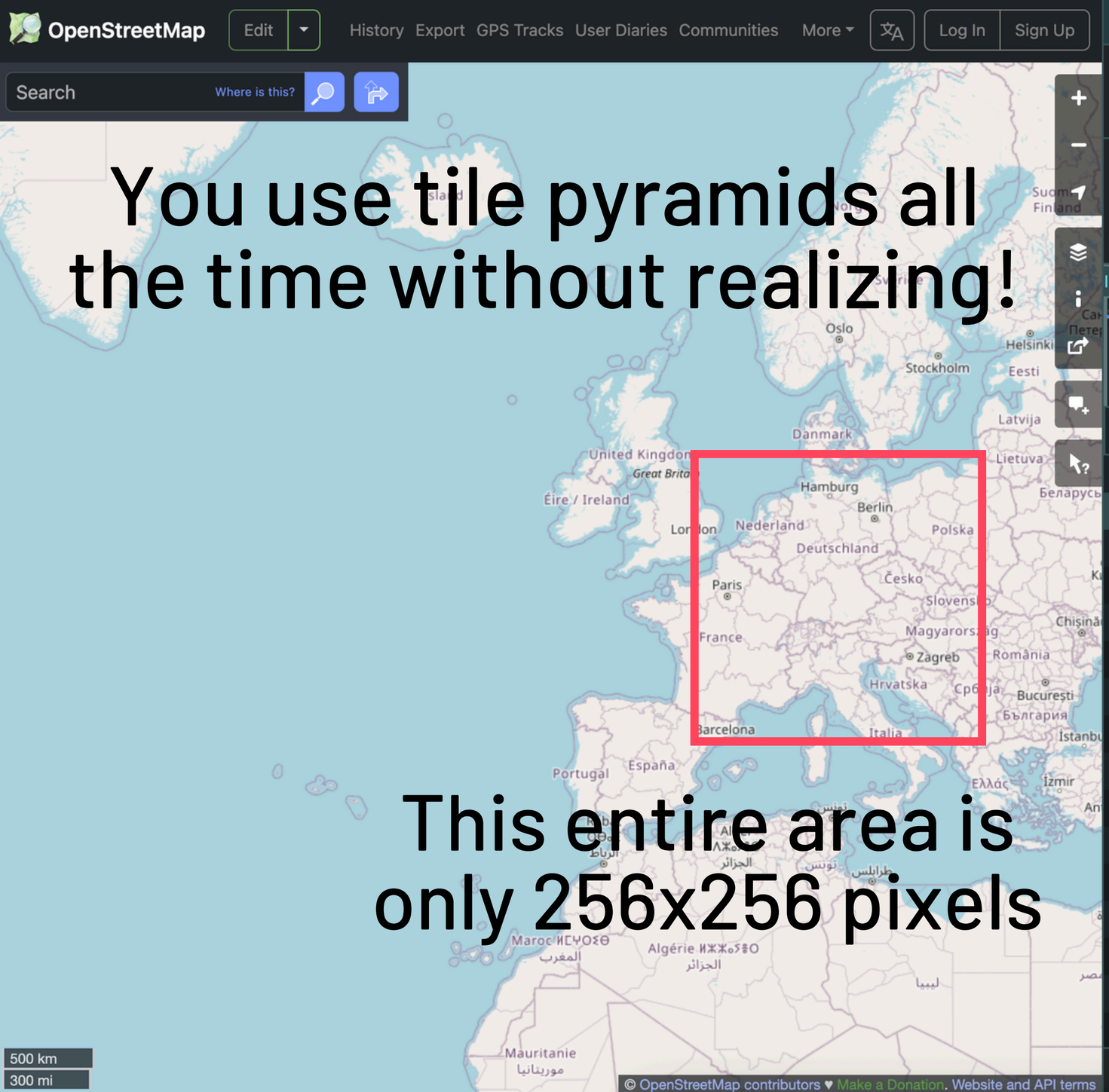


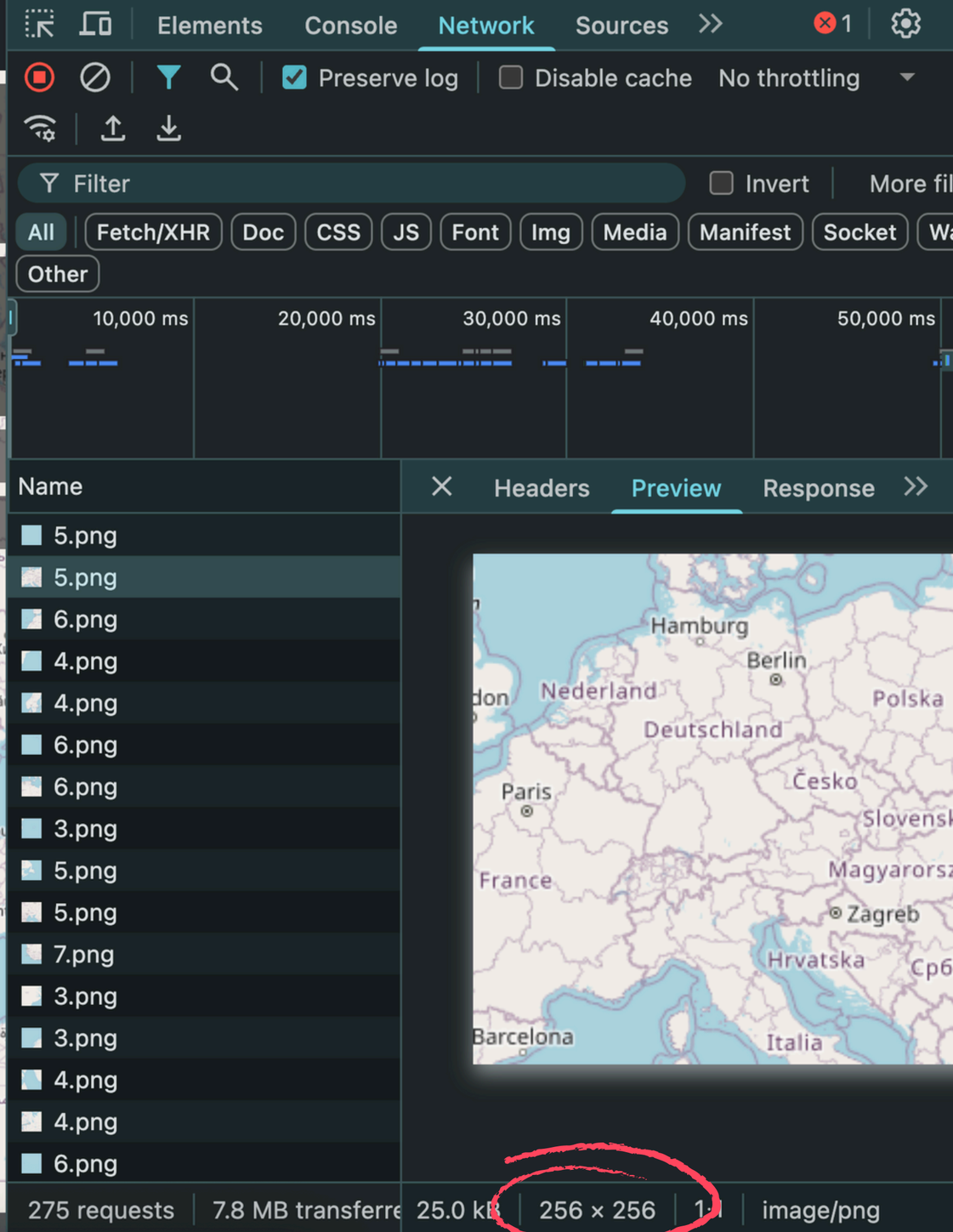
We don’t need to show all the pixels!

Pre-compute **tiles** on several zoom levels

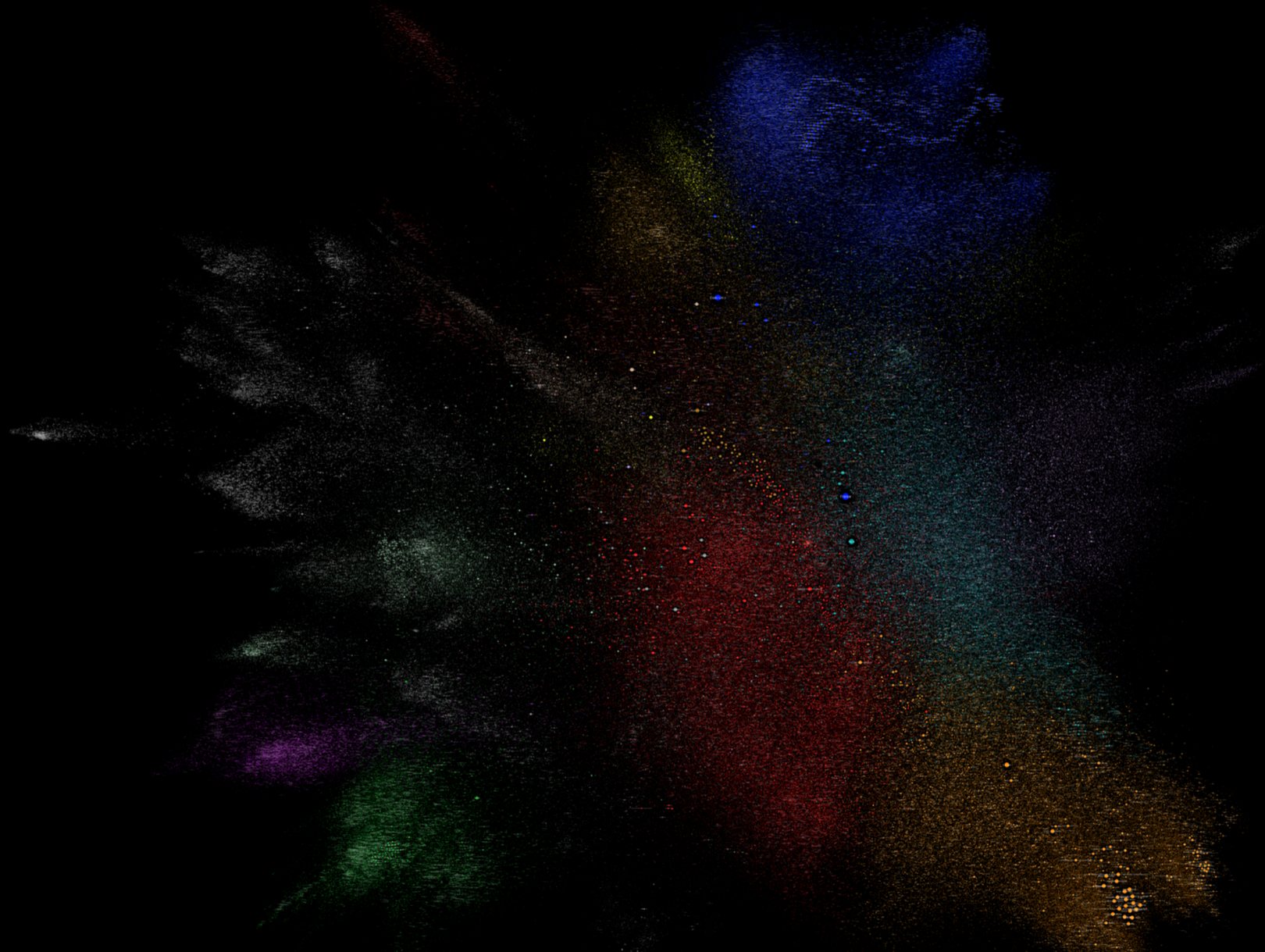
When the user zooms, substitute finer tiles

AGORATLAS


OpenStreetMap
Edit
History
Export
GPS Tracks
User Diaries
Communities
More
Log In
Sign Up
Search
Where is this?
You use tile pyramids all the time without realizing!
This entire area is only 256x256 pixels
Oslo
Stockholm
Helsinki
Eesti
Latvija
Lietuva
Danmark
United Kingdom
Great Britain
Éire / Ireland
Hamburg
Berlin
Nederland
Polska
Deutschland
Česko
Paris
France
Slovensko
Magyarország
Zagreb
Hrvatska
România
București
Barcelona
Italia
España
Portugal
Ελλάς
İzmir
İstanbul
Беларусь
Maroc
Algérie
Mauritanie
500 km
300 mi
© OpenStreetMap contributors ♥ Make a Donation. Website and API terms

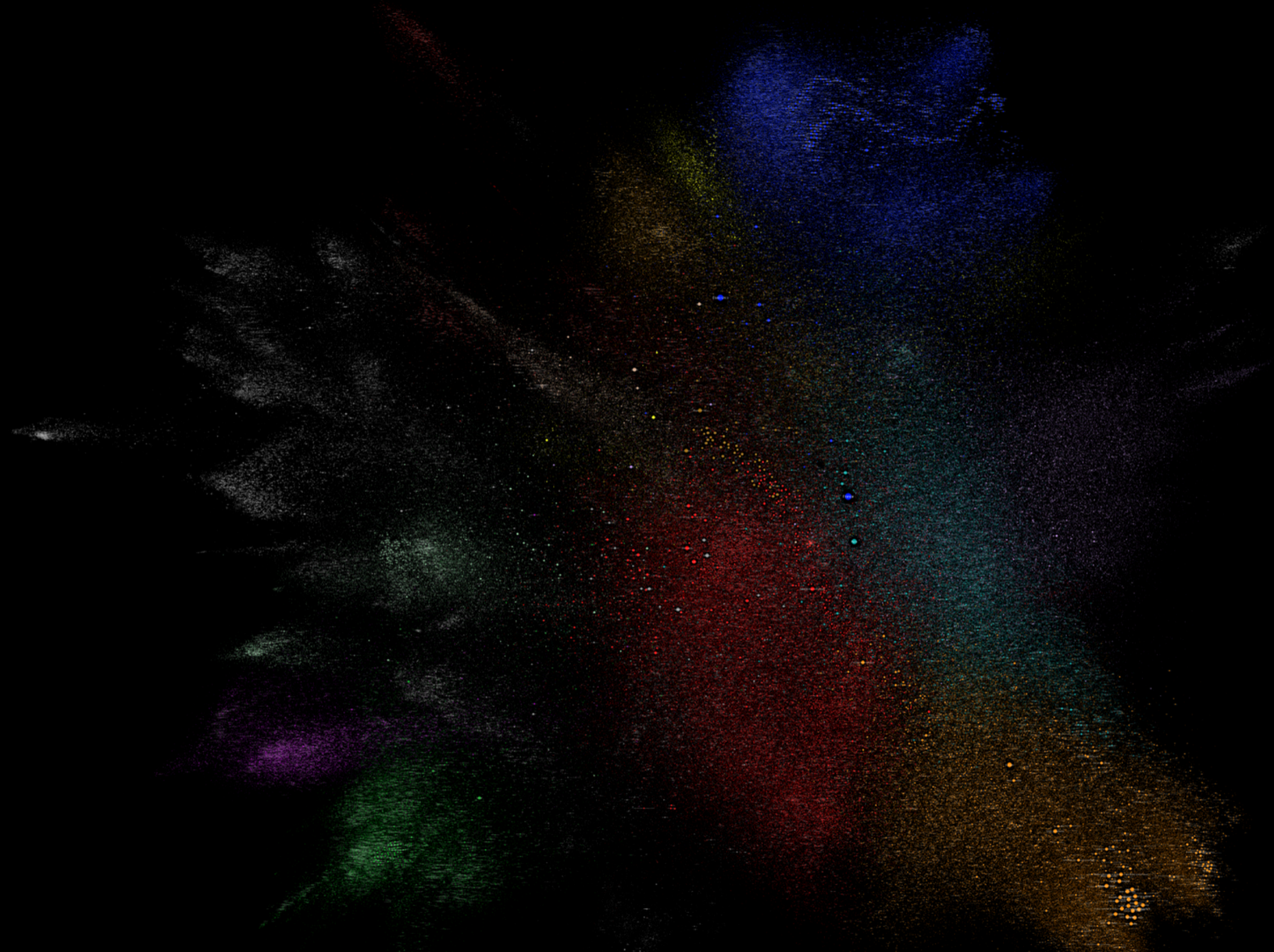

Elements
Console
Network
Sources
Preserve log
Disable cache
No throttling
Filter
Invert
All
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Manifest
Socket
Other
10,000 ms
20,000 ms
30,000 ms
40,000 ms
50,000 ms
Name
5.png
5.png
6.png
4.png
4.png
6.png
6.png
3.png
5.png
5.png
7.png
3.png
3.png
4.png
4.png
6.png
Headers
Preview
Response
Hamburg
Berlin
Nederland
Polska
Deutschland
Česko
Paris
France
Zagreb
Hrvatska
Barcelona
Italia
275 requests
7.8 MB transferre
25.0 kB
256 × 256
image/png

Let’s see it in action!



AGORATLAS

Let’s see it in action!



graph.agoratlas.com/image/wiki

AGORATLAS

Thank you!

Mathis Hammel
mathis@agoratlas.com
CTO, Agoratlas

Microsoft® MVP Most Valuable Professional

Google Developer Expert

AGORATLAS

Social data agency

- Foreign inference tracking
- Marketing strategy
- PR crisis response