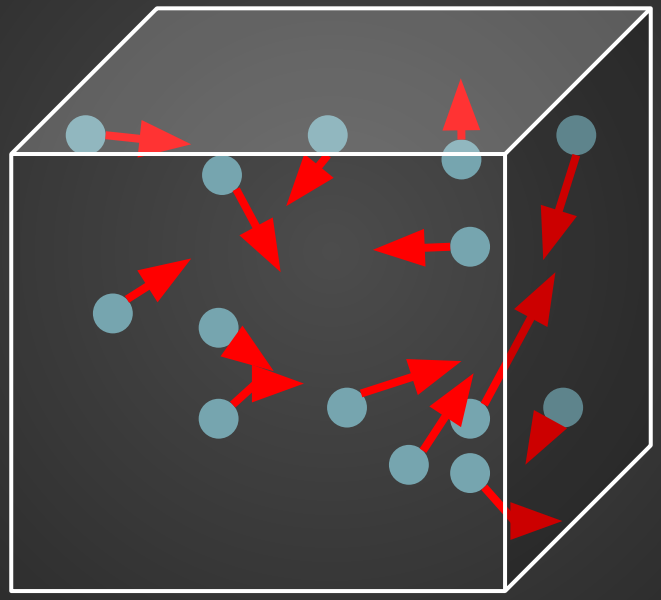
# Volumes with yt



Matthew Turk
Columbia University

“Exascale is made of people!”

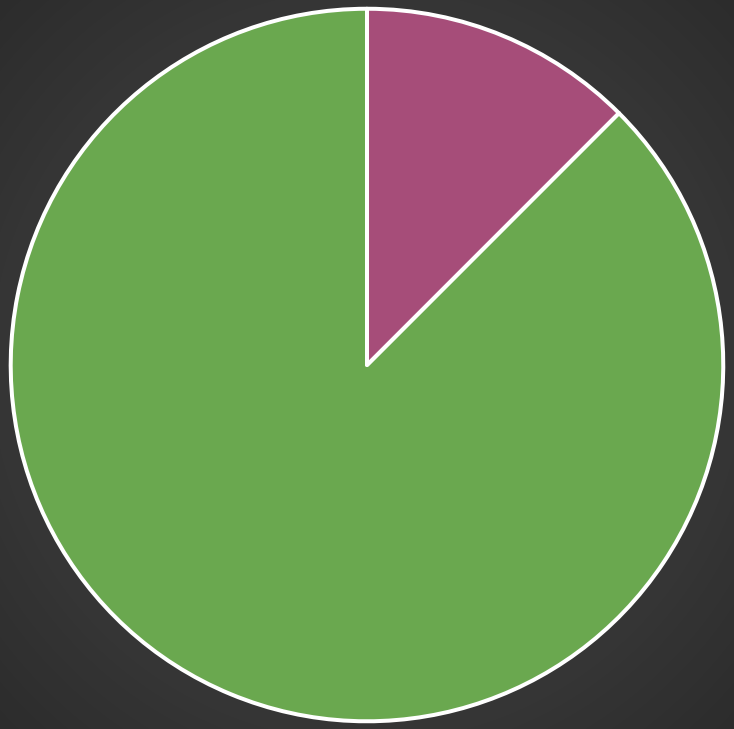
- Mike Warren, 2014

# Scientific Inquiry

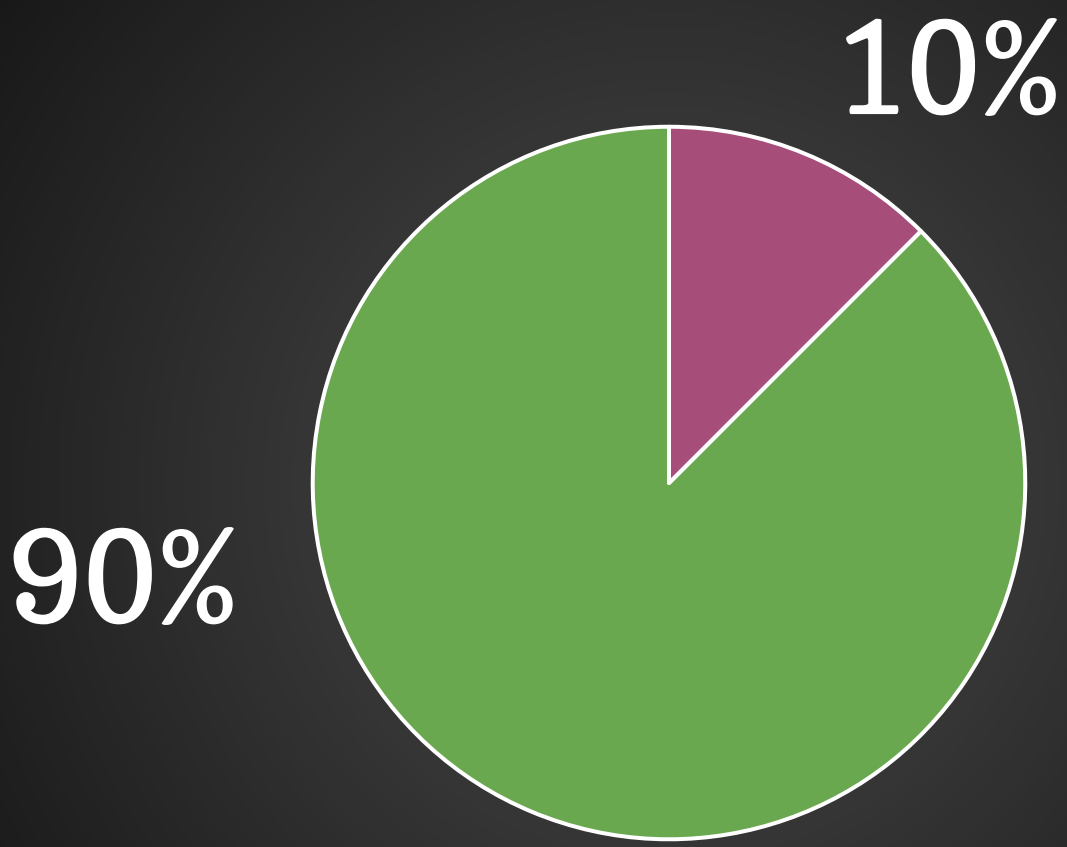
# Analyzing Data

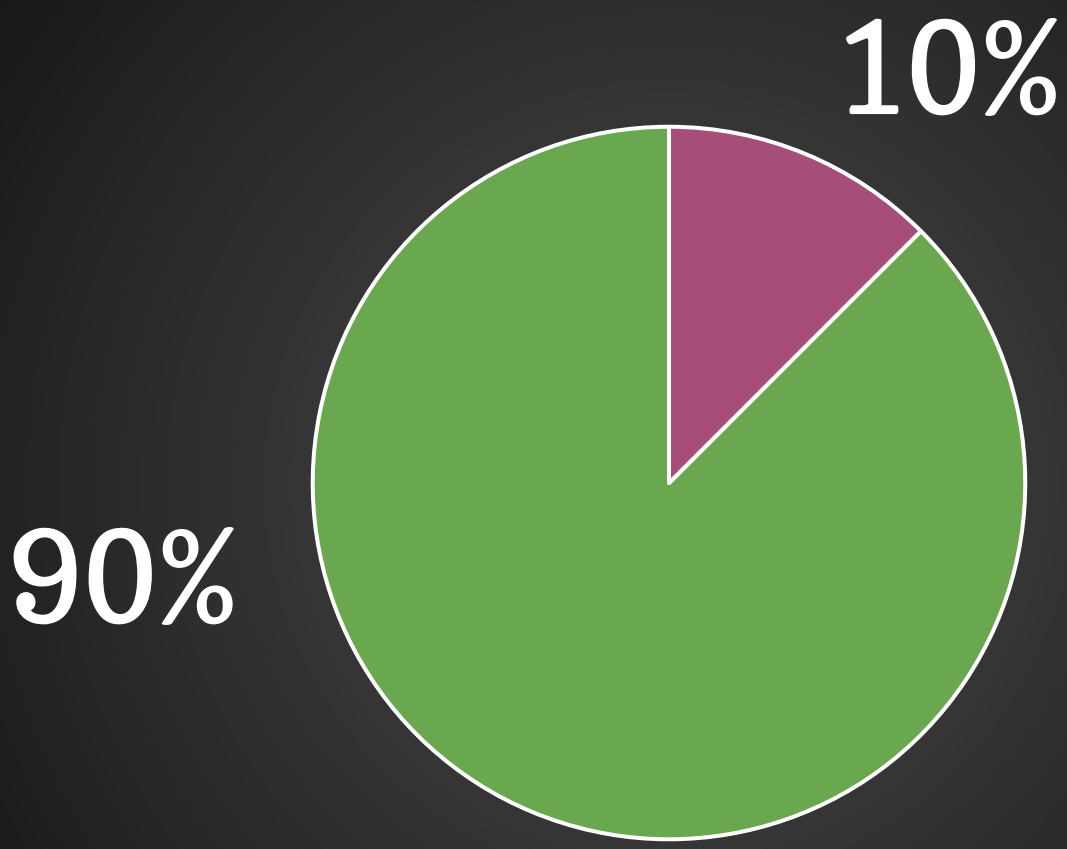
# Running Simulations

Run



Study

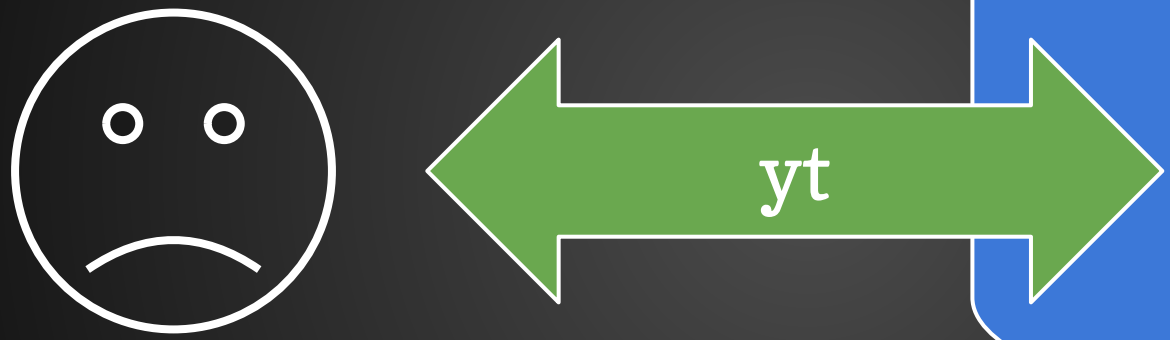

10%
90%

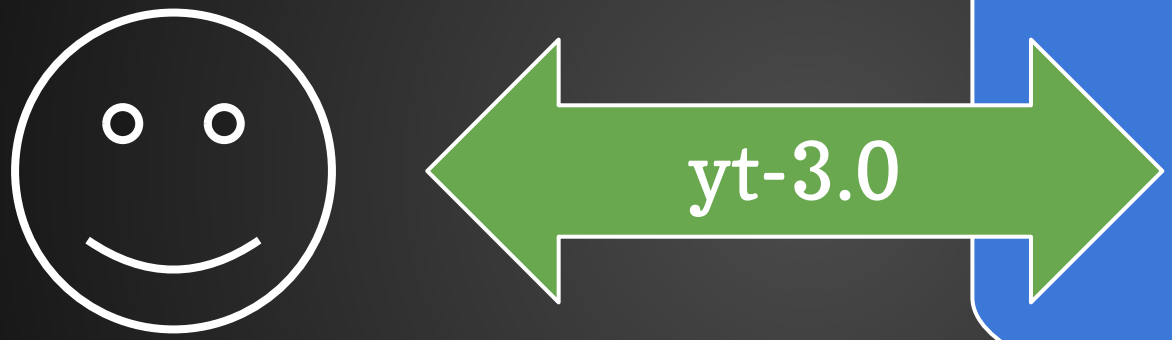



Dekel's Law


Data


yt
Data


yt

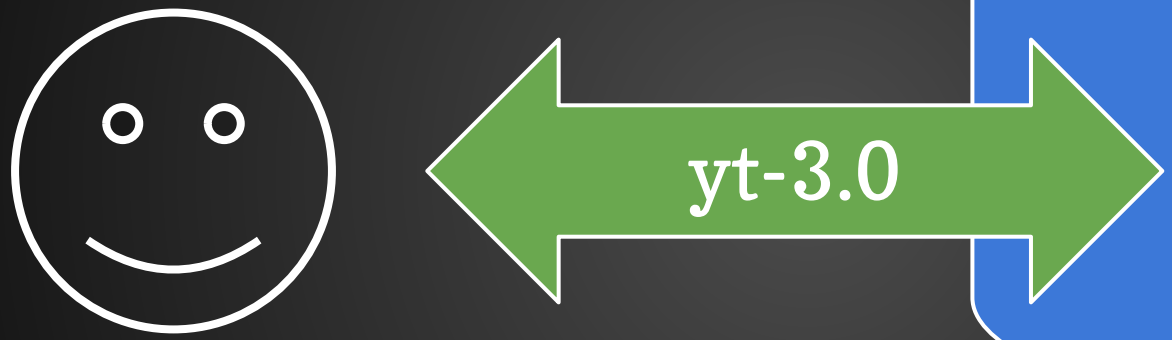

yt-3.0


yt-3.0

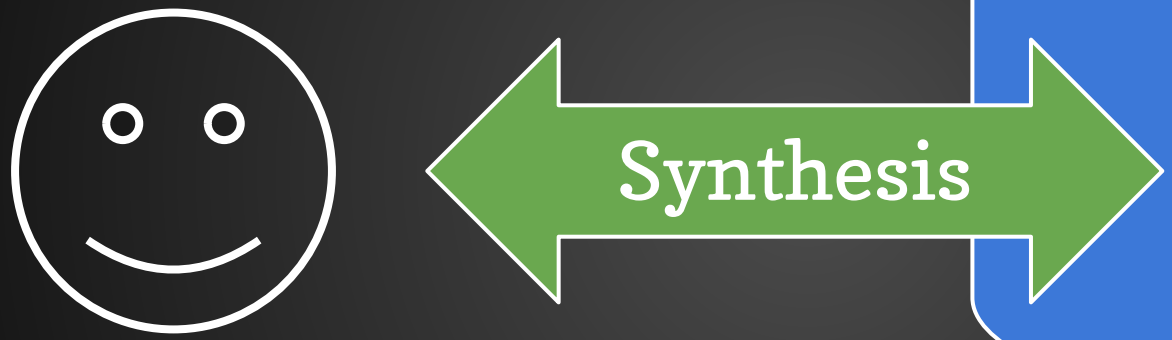

We got to break,
then fix.

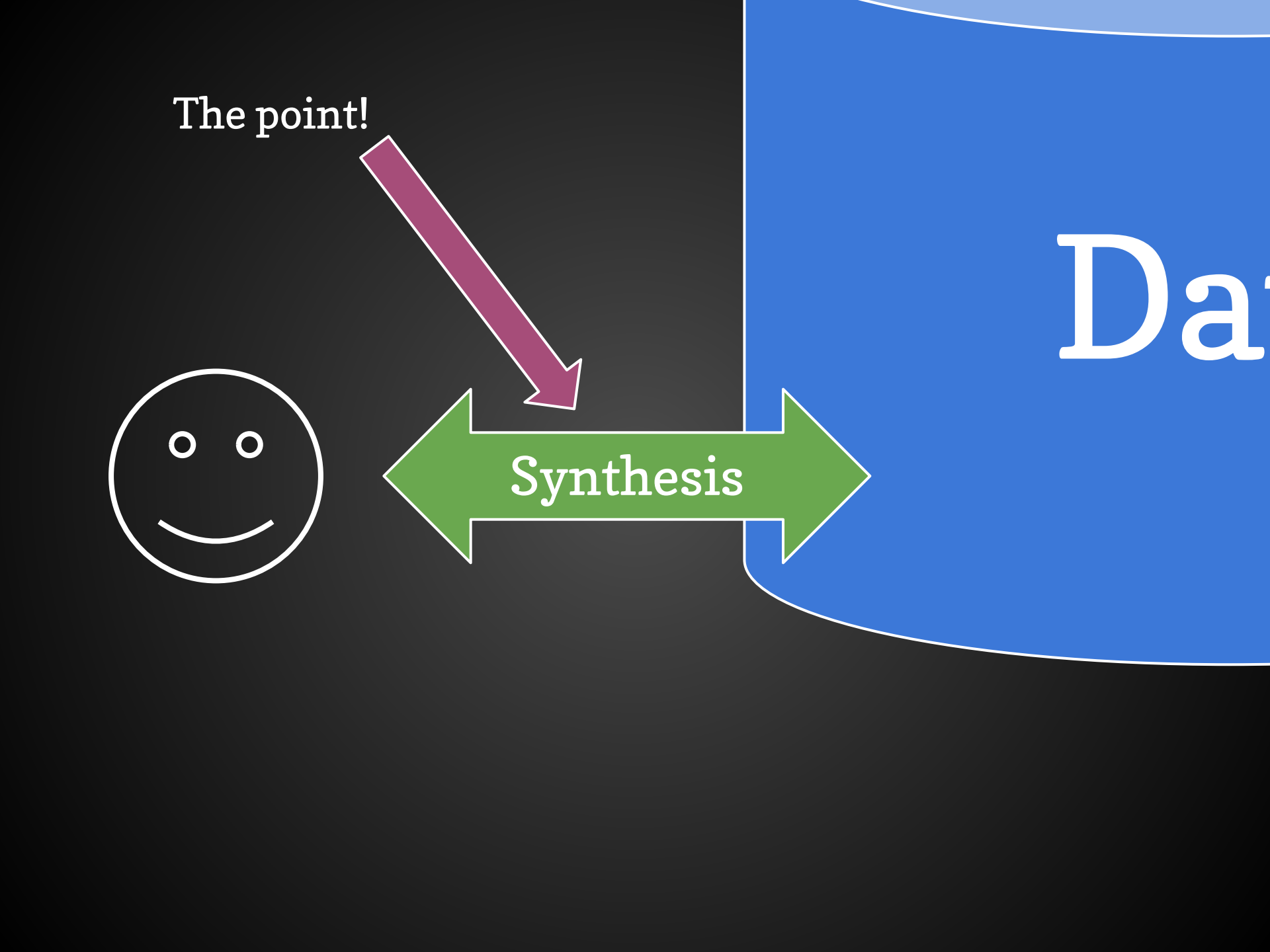
# yt-3.0

SPH, octrees, units, field types, ...


yt-3.0

yt is not the point.


Synthesis


The point!
Synthesis
Da

# How do we get there?

Clarity

Abstraction

Delegation

# Develop a data language.

# Develop a data language.

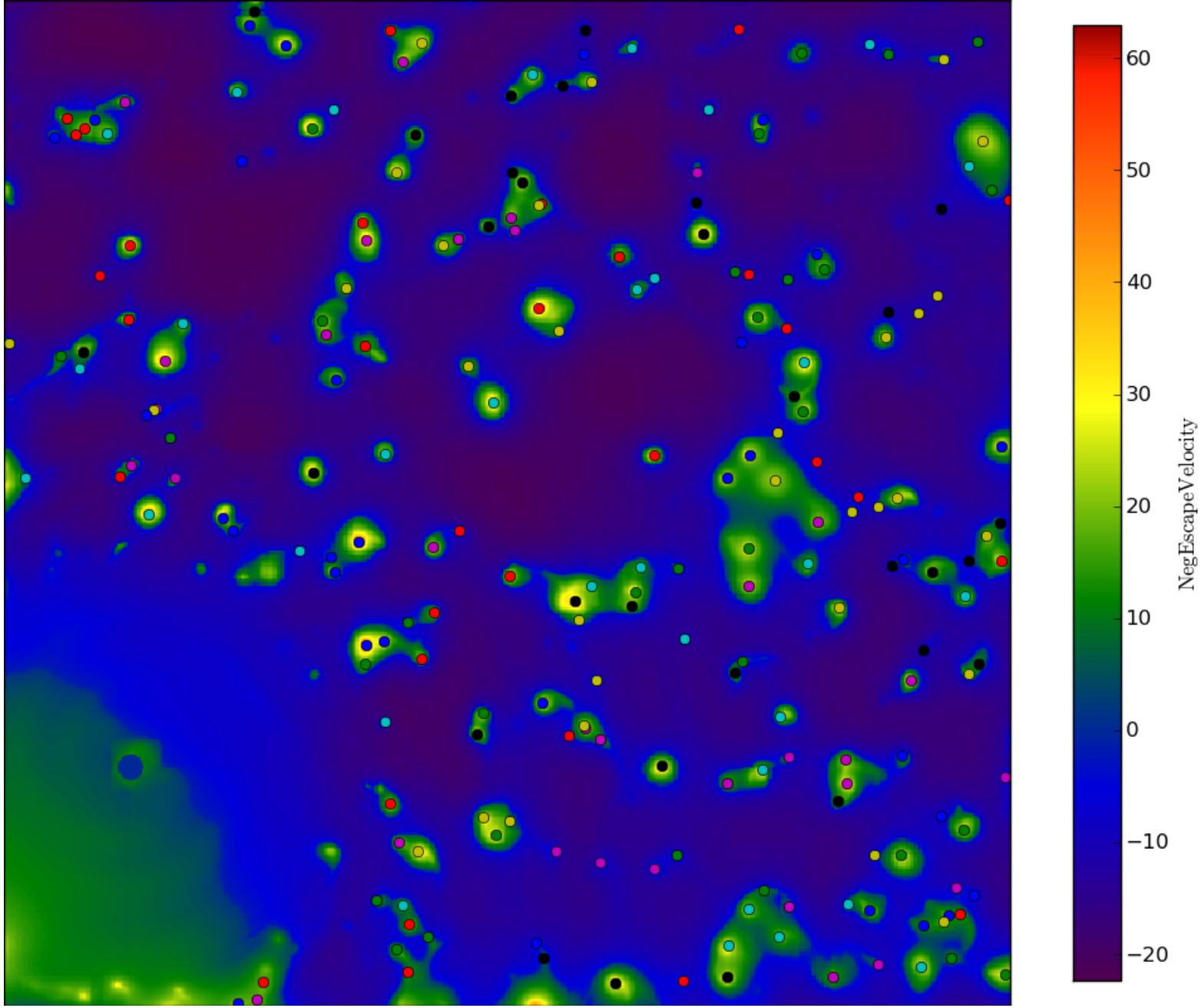
Inspection, Analysis, & Visualization

Inspection, Analysis, & Visualization

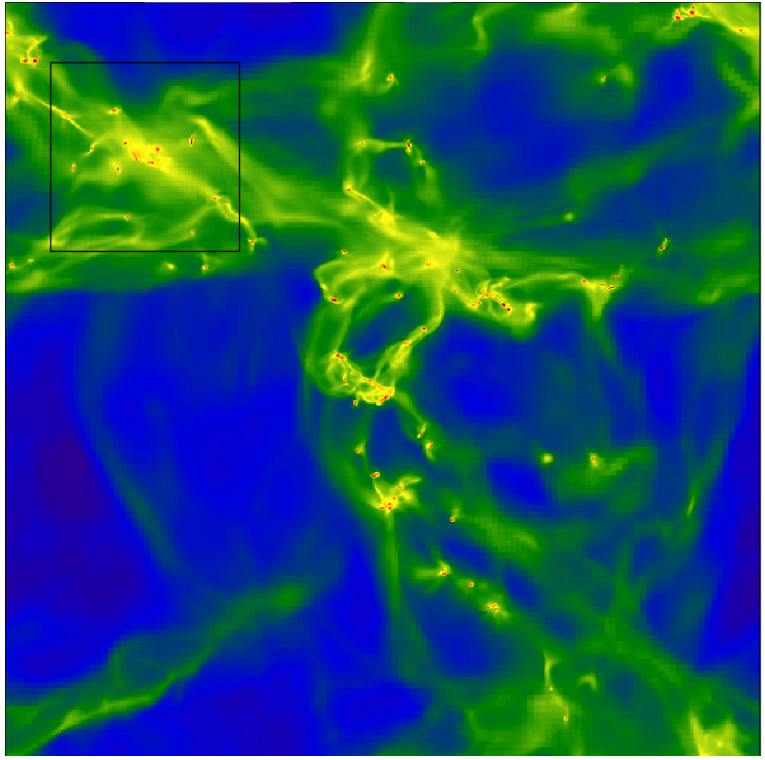
```
import yt
ds = yt.load("output_00084/info_00084.txt")
sp = ds.sphere("max", (1.0, "Mpc"))
print sp["density"]

[  3.713965e-32, ..., 8.940313e-30] g/cm**3
```

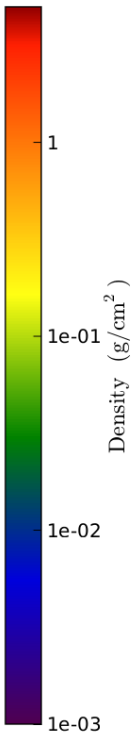
```
import yt
ds = yt.load("output_00084/info_00084.txt")
sp = ds.sphere("max", (1.0, "Mpc"))
sp.quantities.extrema("density")

(3.196587e-32 g/cm**3, 7.158091e-24 g/cm**3)
```

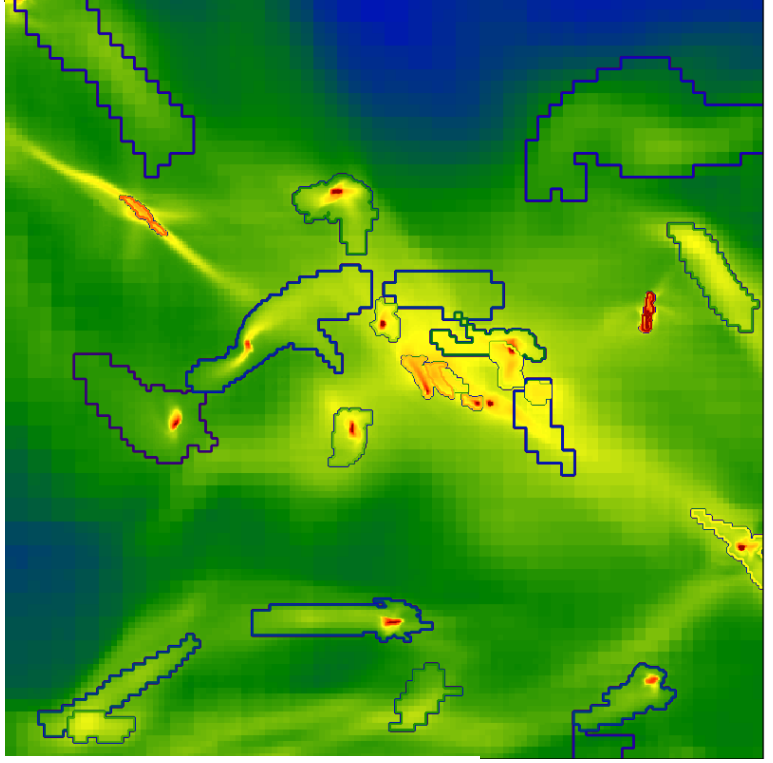
Inspection, Analysis, & Visualization

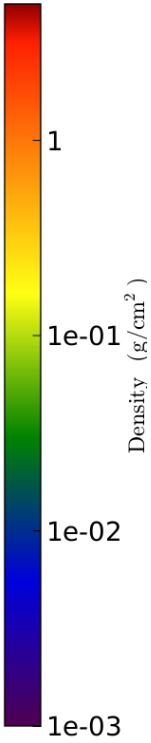


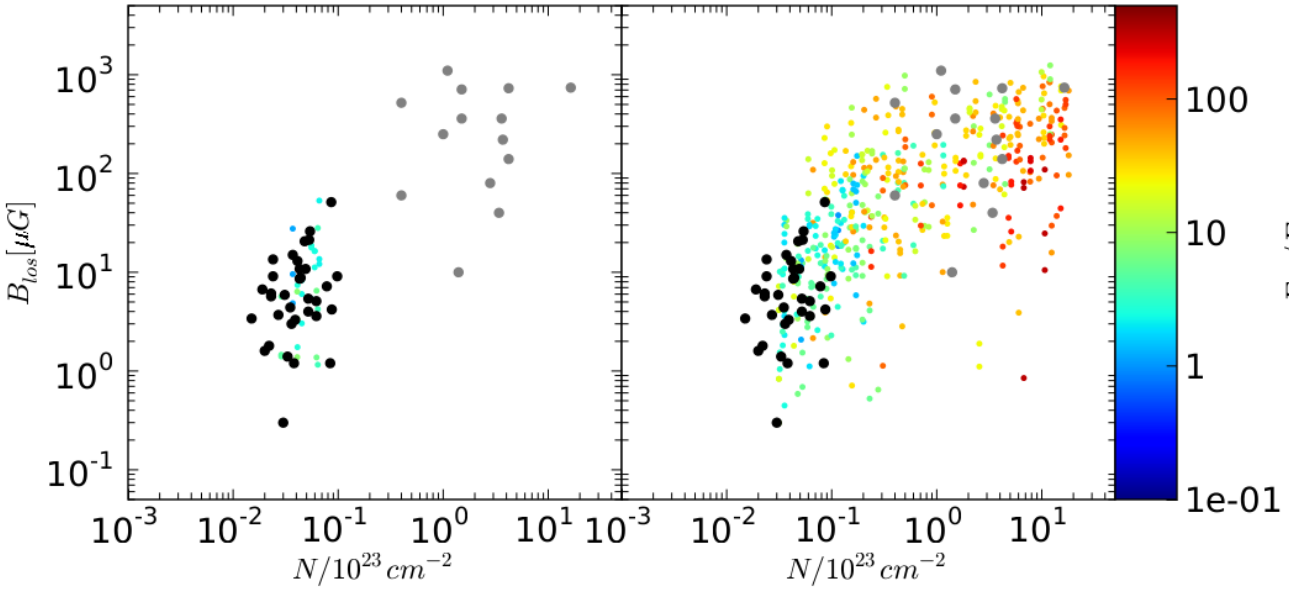

Tasker et al 2013



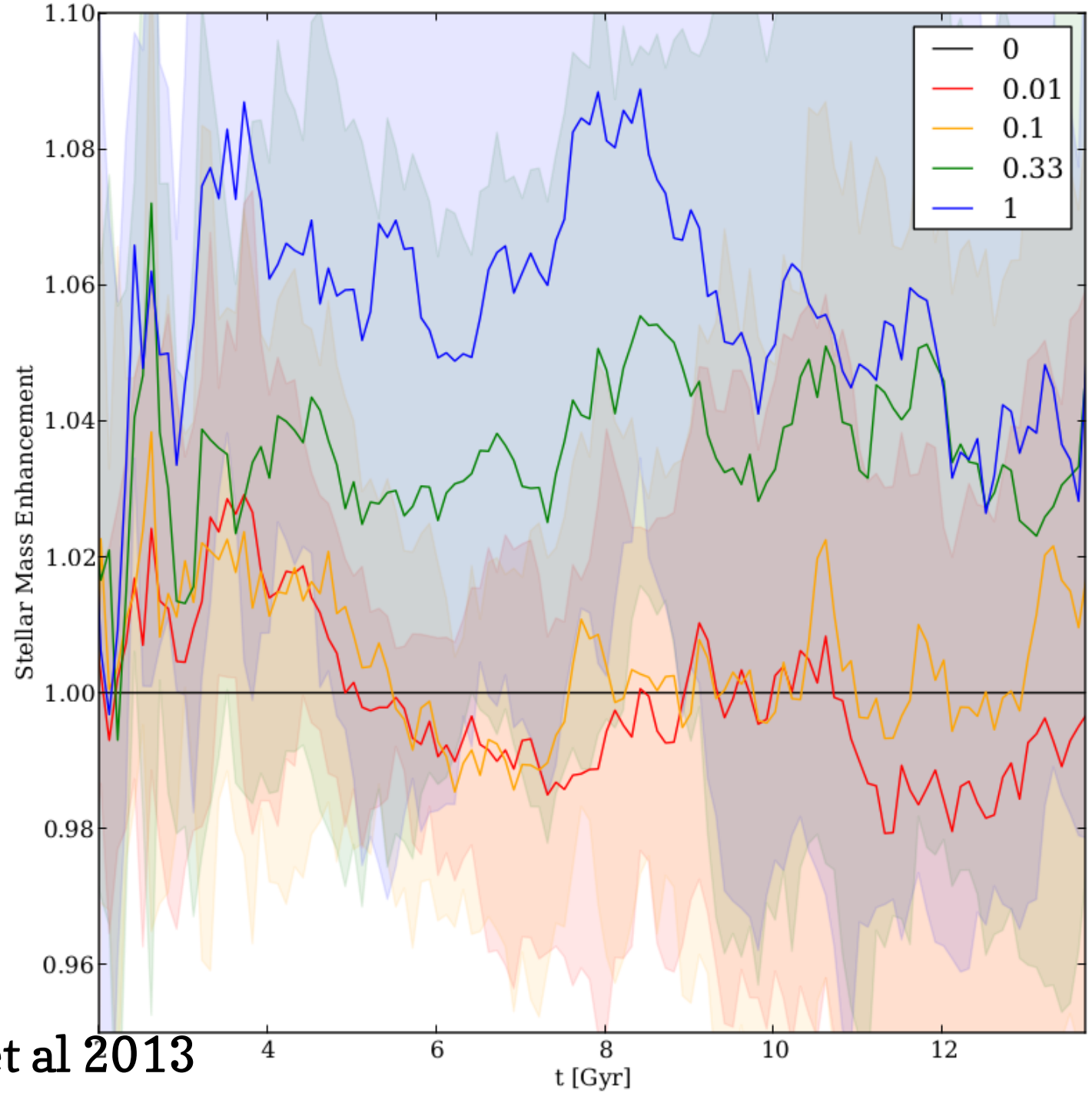


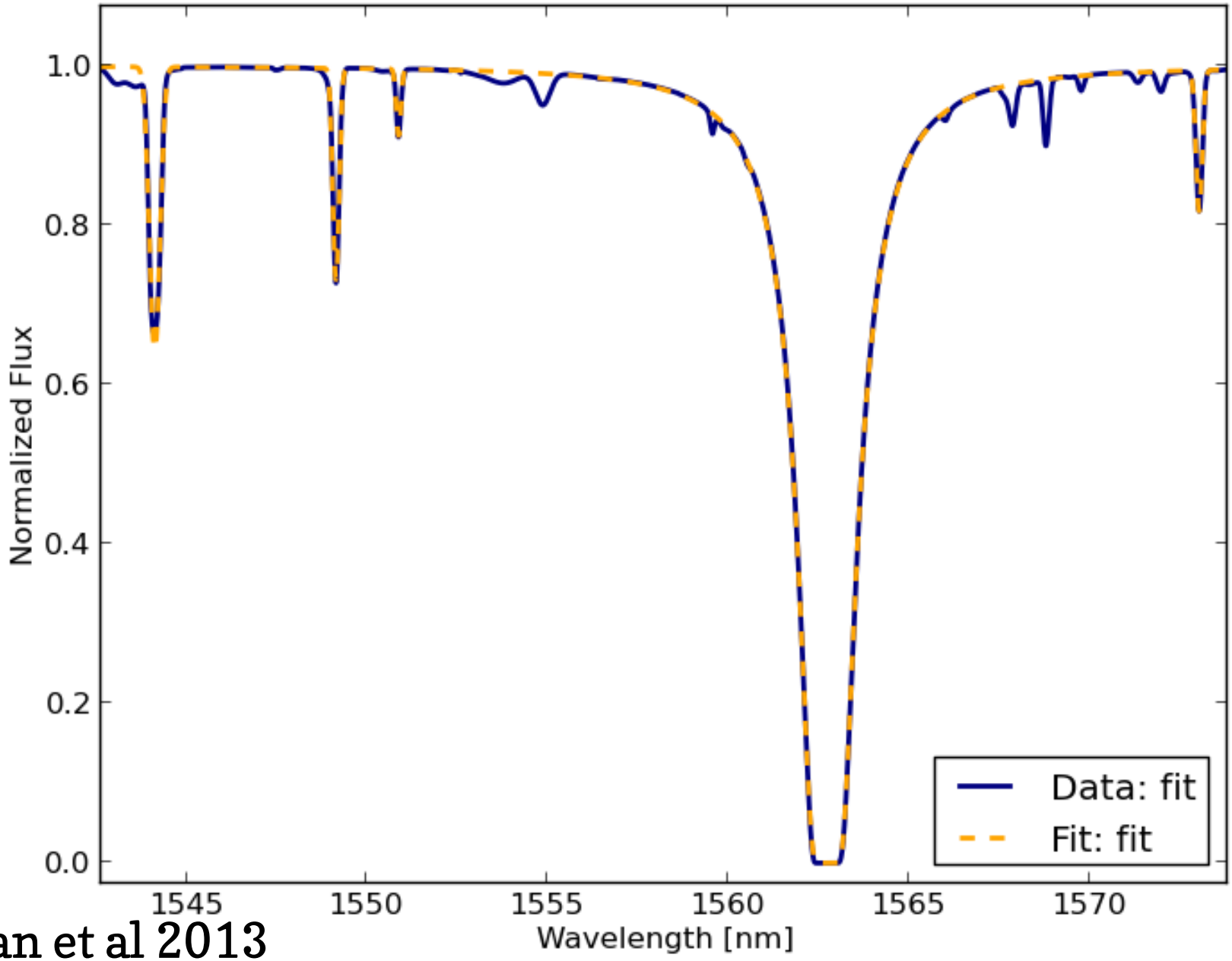


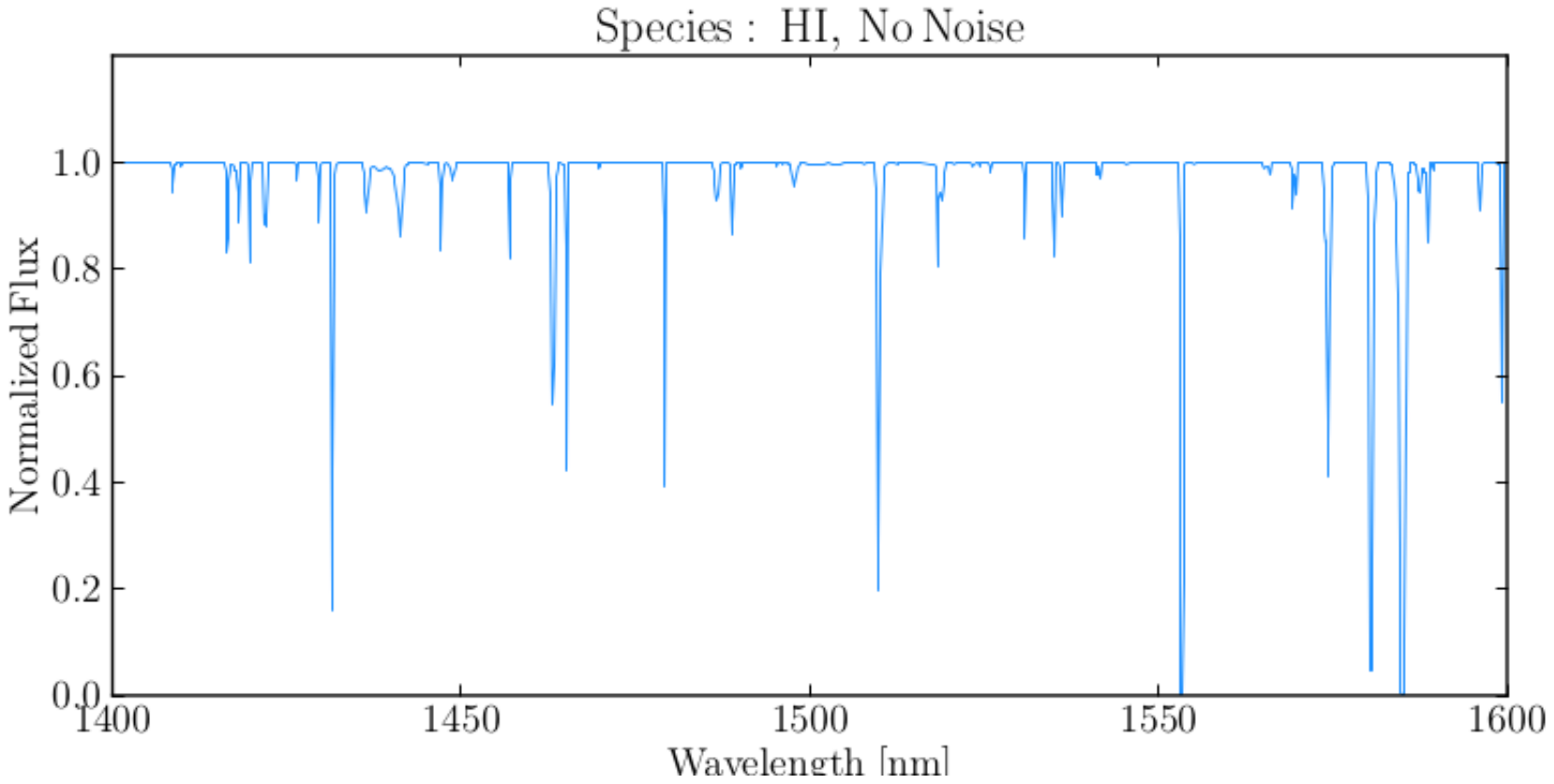

Collins et al 2012



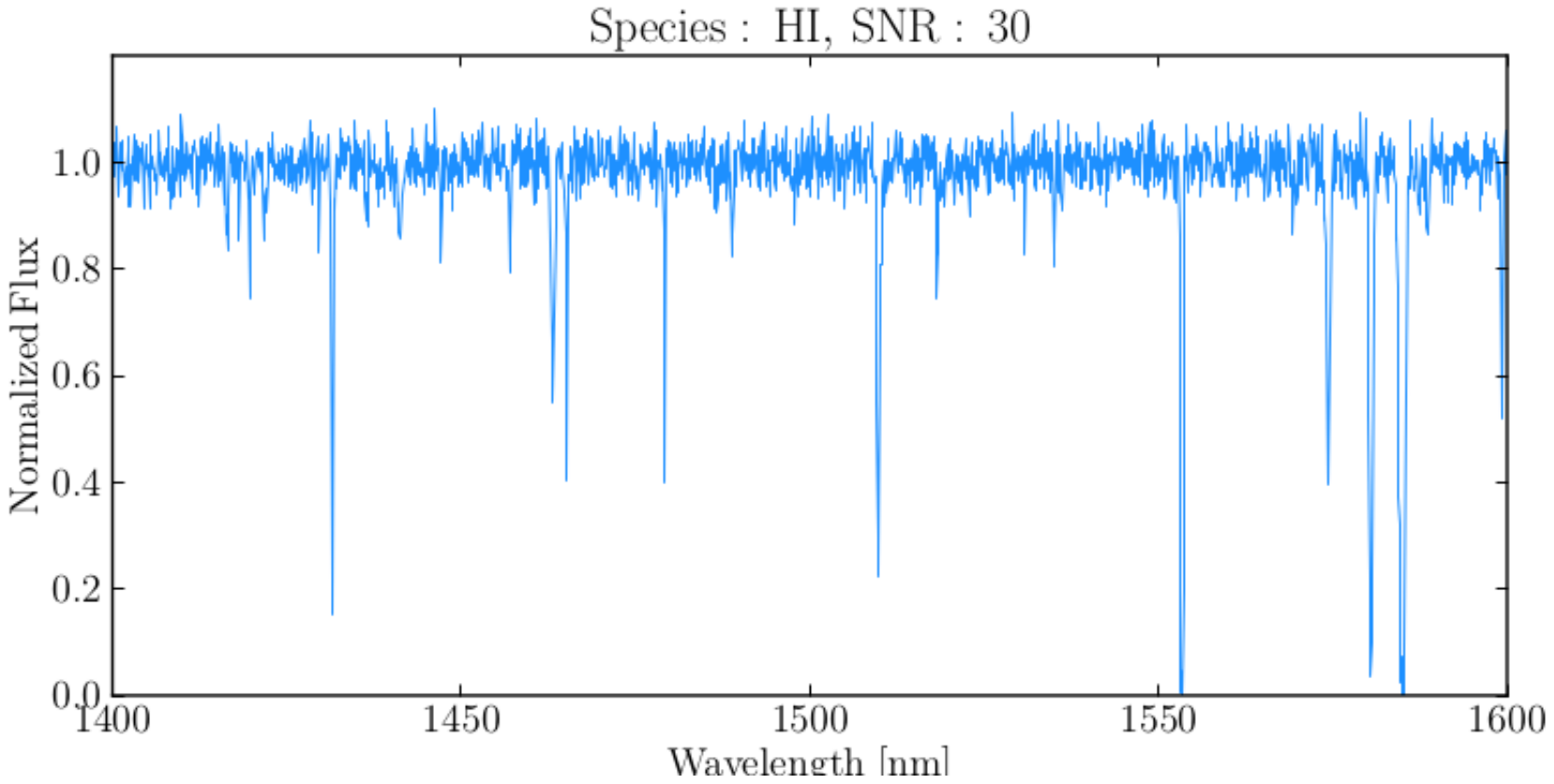

Smith et al 2013



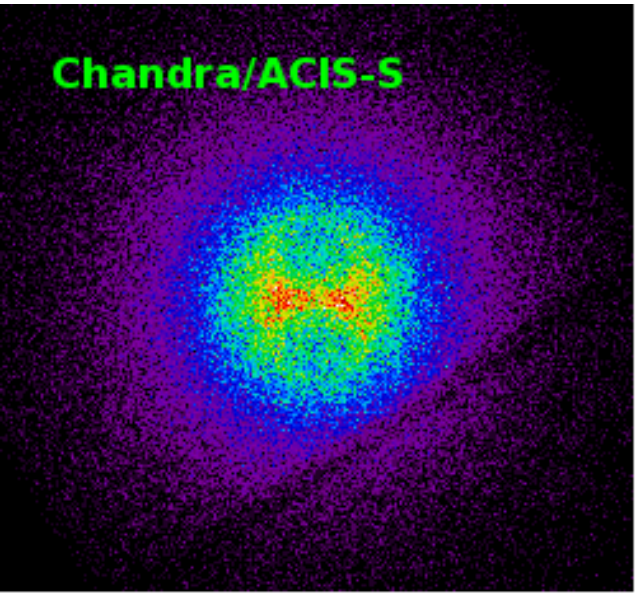

Egan et al 2013

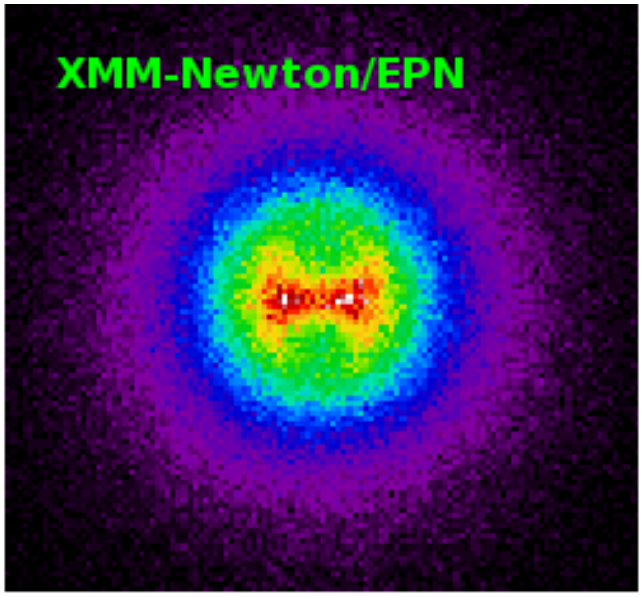


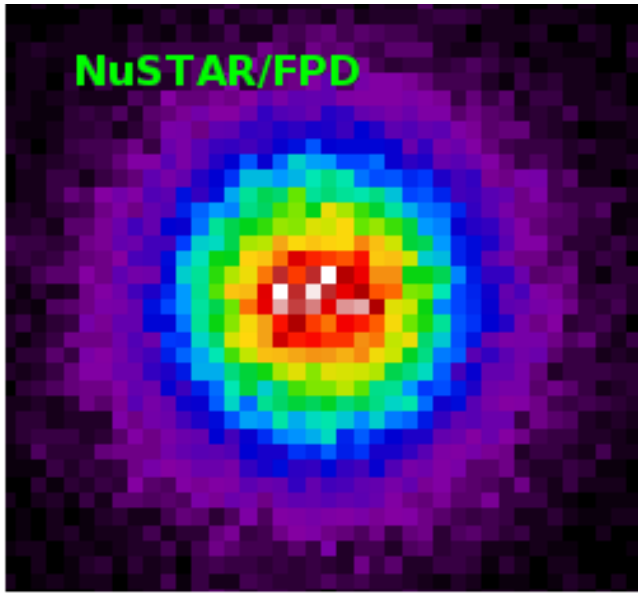

Egan et al 2013



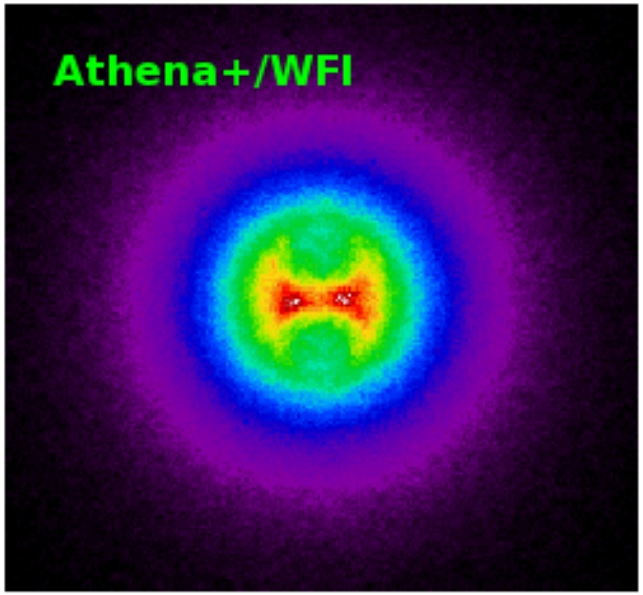

Egan et al 2013



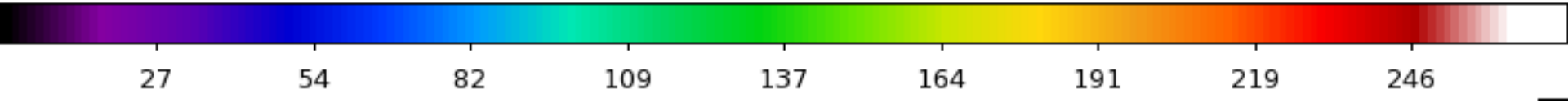




Astro-H/SXS

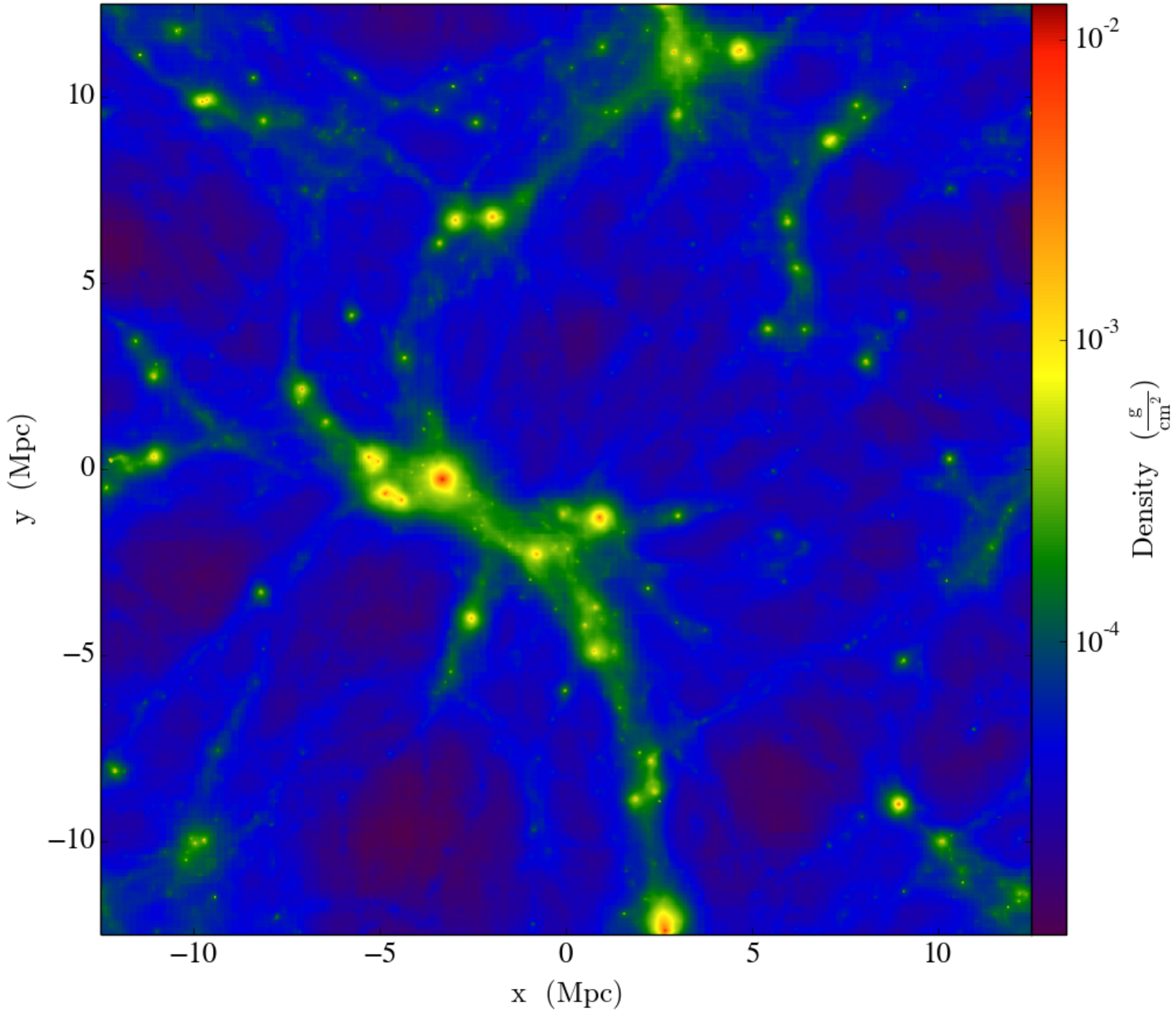



ZuHone et al

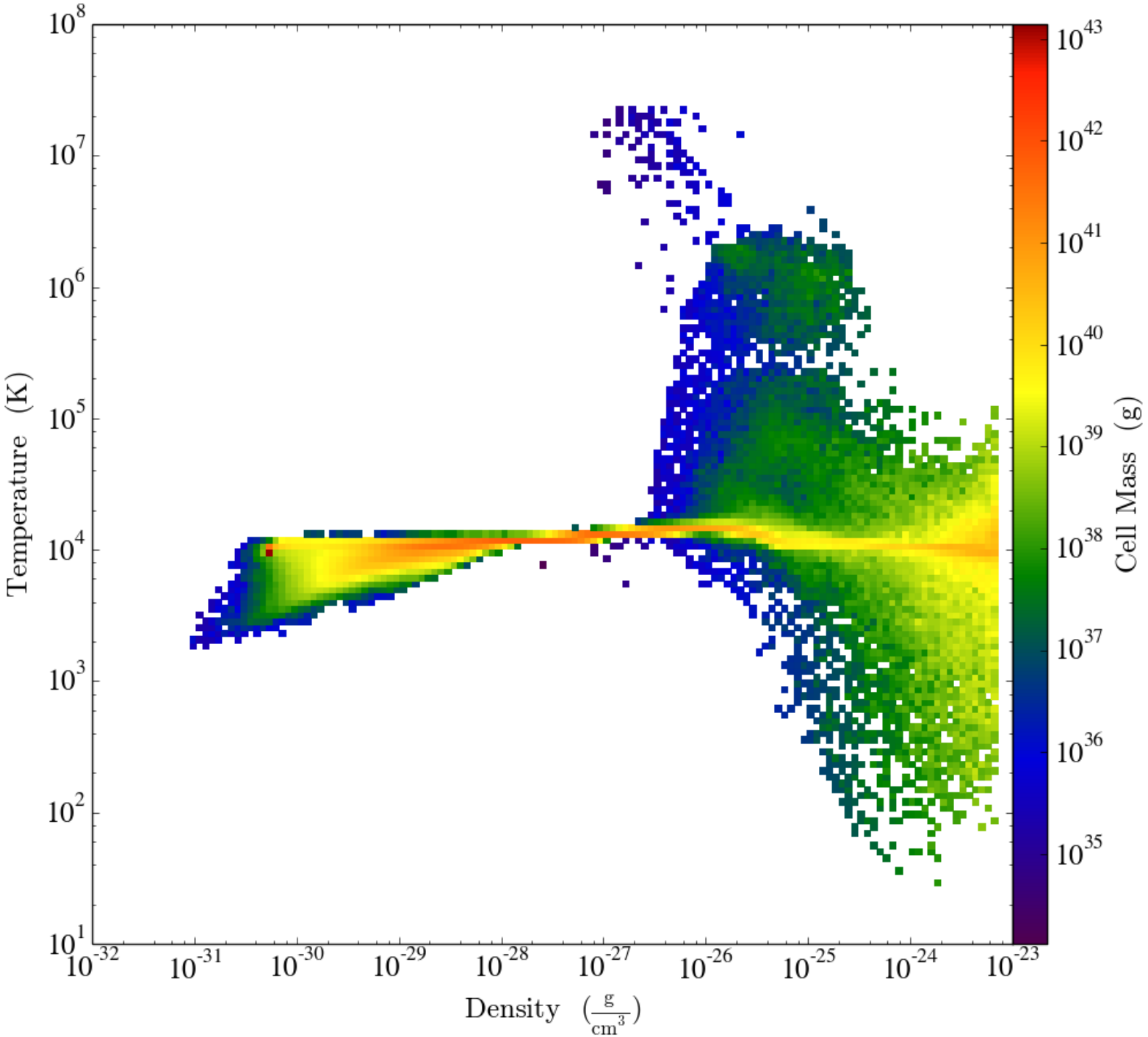


# Inspection, Analysis, & Visualization

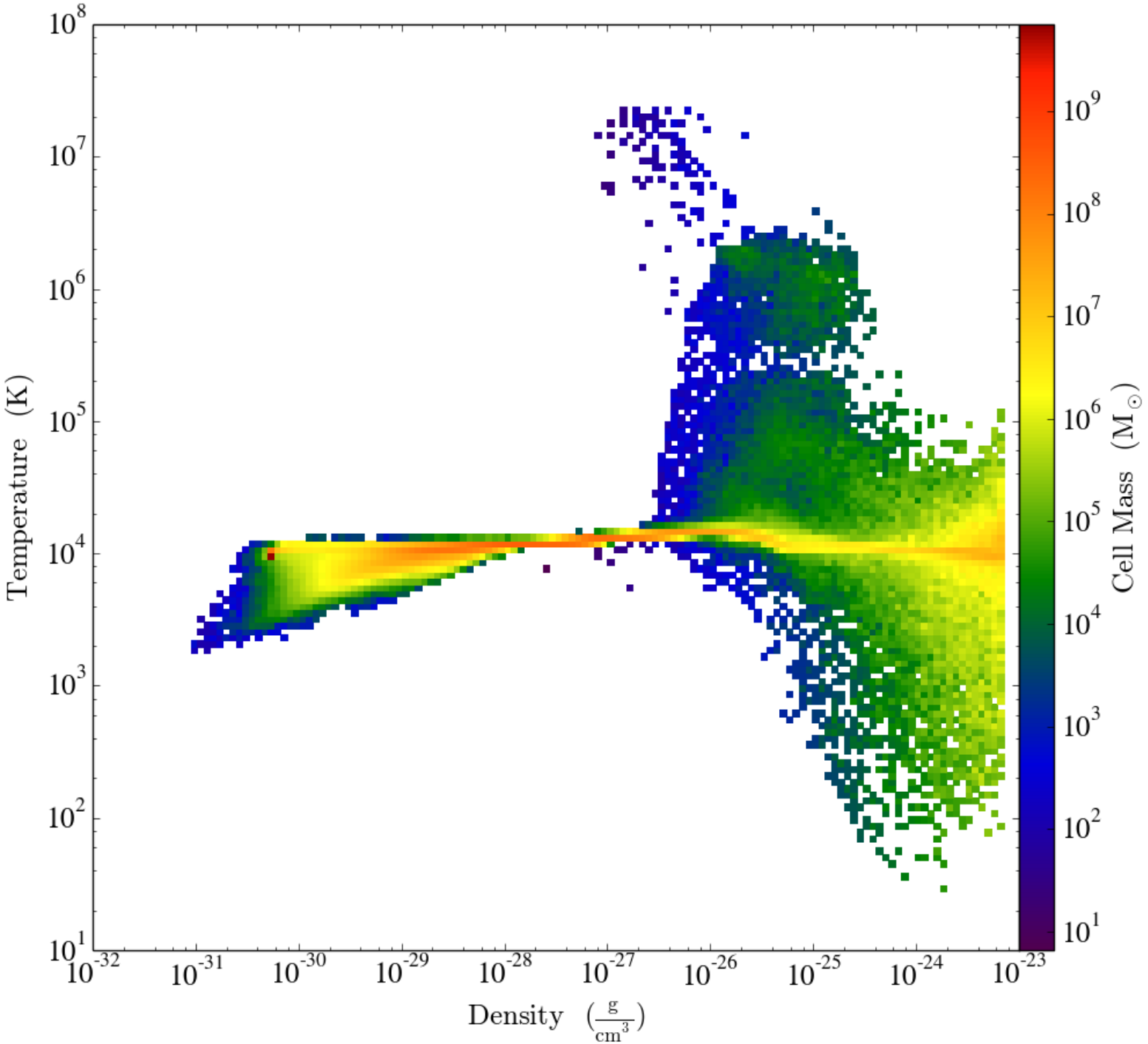
```
import yt
ds = yt.load("snapshot_033/snap_033.0.hdf")
p = yt.ProjectionPlot(ds, "x", "density")
p.save()
```

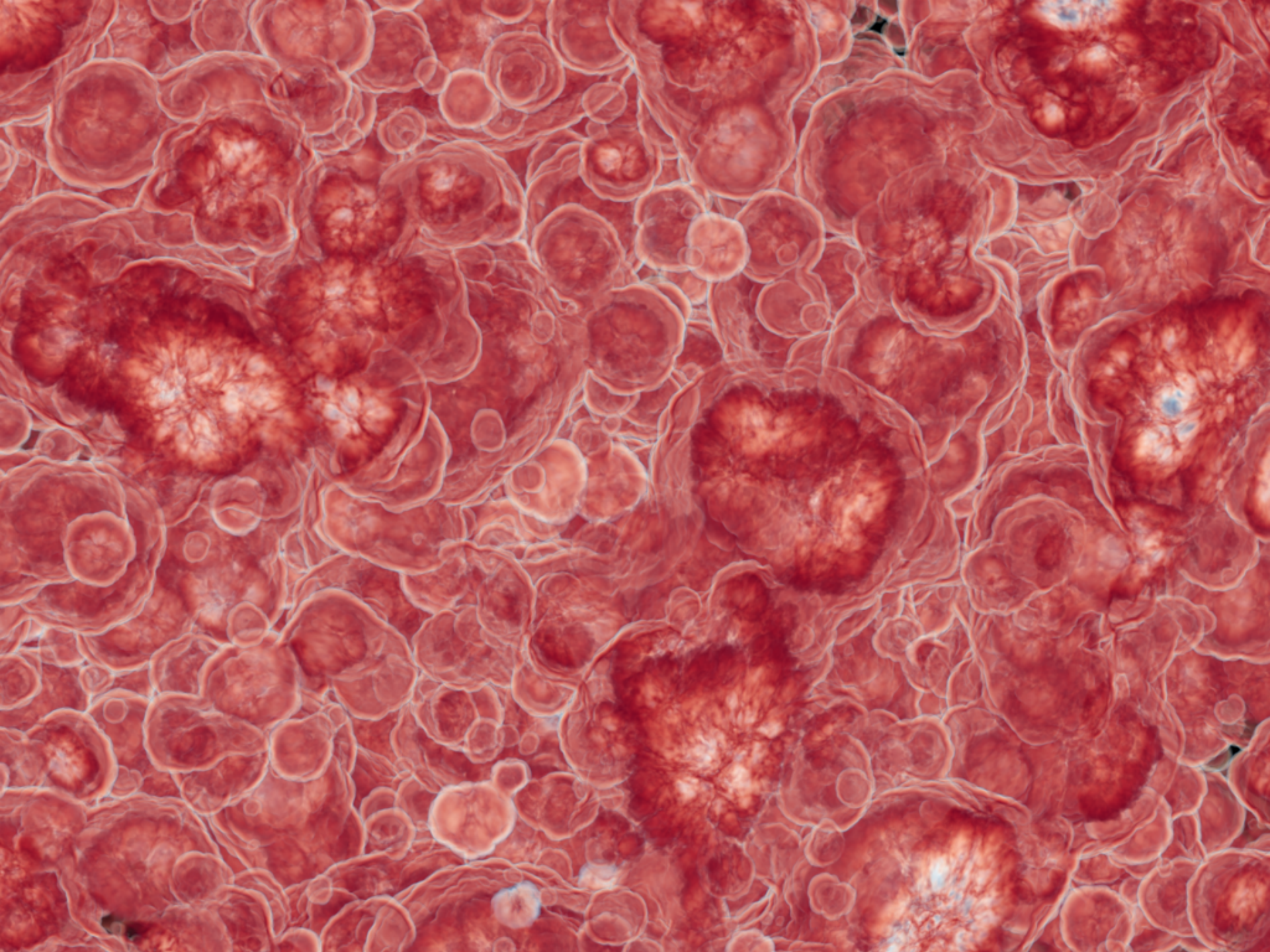

10
5
0
−5
−10
y (Mpc)
−10
−5
0
5
10
x (Mpc)
$10^{-2}$
$10^{-3}$
$10^{-4}$
Density $\left(\frac{\mathrm{g}}{\mathrm{cm}^2}\right)$

```
import yt
ds = yt.load("galaxy0030/galaxy0030")
ad = ds.all_data()
p = yt.PhasePlot(ad, "density",
        "temperature", "cell_mass", None)
p.save()
```


Temperature (K)
$10^{8}$
$10^{7}$
$10^{6}$
$10^{5}$
$10^{4}$
$10^{3}$
$10^{2}$
$10^{1}$
$10^{-32}$
$10^{-31}$
$10^{-30}$
$10^{-29}$
$10^{-28}$
$10^{-27}$
$10^{-26}$
$10^{-25}$
$10^{-24}$
$10^{-23}$
Density $(\frac{g}{cm^3})$
Cell Mass (g)
$10^{43}$
$10^{42}$
$10^{41}$
$10^{40}$
$10^{39}$
$10^{38}$
$10^{37}$
$10^{36}$
$10^{35}$

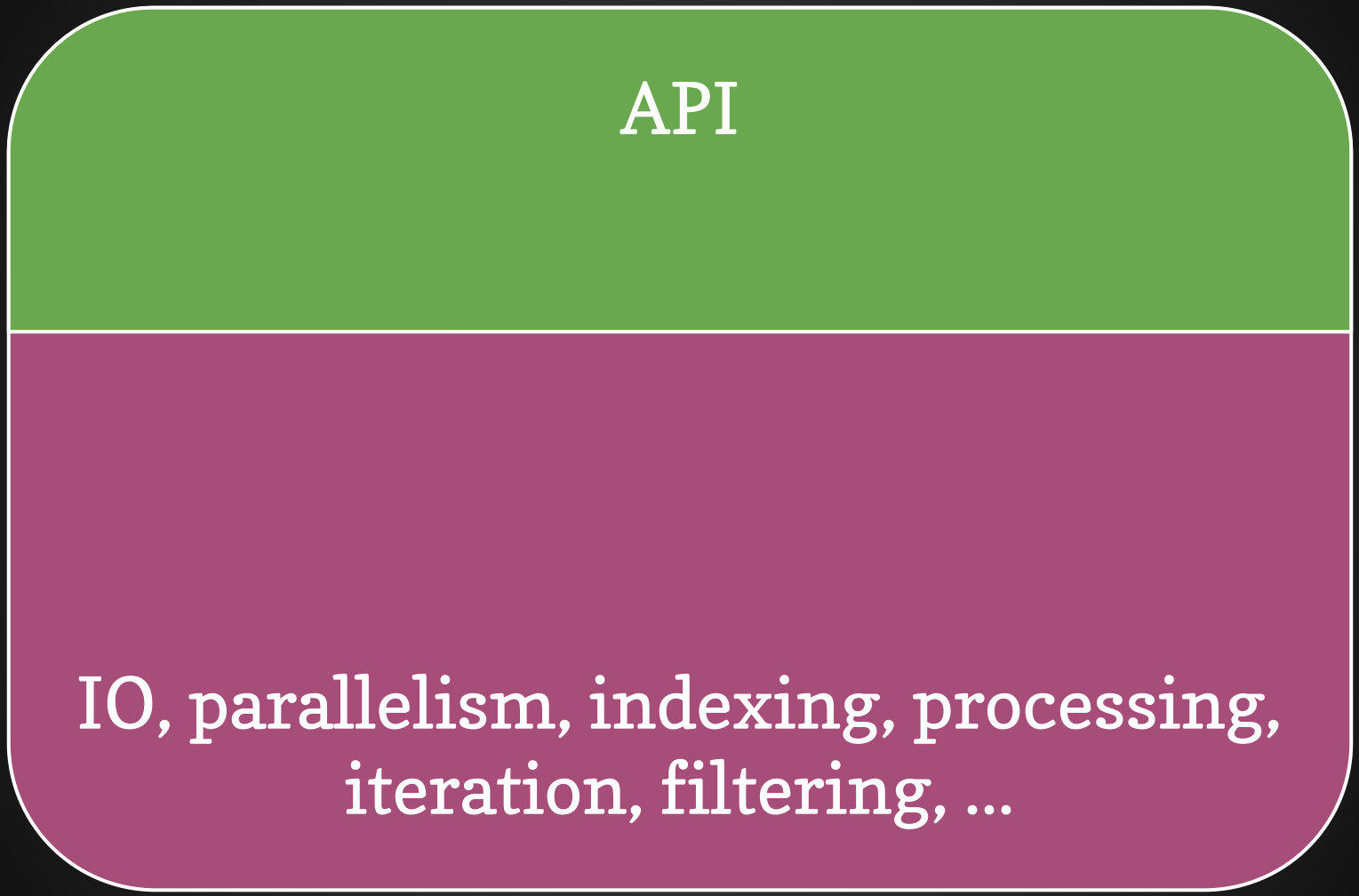
```
import yt
ds = yt.load("galaxy0030/galaxy0030")
ad = ds.all_data()
p = yt.PhasePlot(ad, "density",
        "temperature", "cell_mass", None)
p.set_unit("cell_mass", "Msun")
p.save()
```


Temperature (K)
$10^{8}$
$10^{7}$
$10^{6}$
$10^{5}$
$10^{4}$
$10^{3}$
$10^{2}$
$10^{1}$
$10^{-32}$
$10^{-31}$
$10^{-30}$
$10^{-29}$
$10^{-28}$
$10^{-27}$
$10^{-26}$
$10^{-25}$
$10^{-24}$
$10^{-23}$
Density $(\frac{g}{cm^3})$
Cell Mass $(M_\odot)$
$10^{9}$
$10^{8}$
$10^{7}$
$10^{6}$
$10^{5}$
$10^{4}$
$10^{3}$
$10^{2}$
$10^{1}$

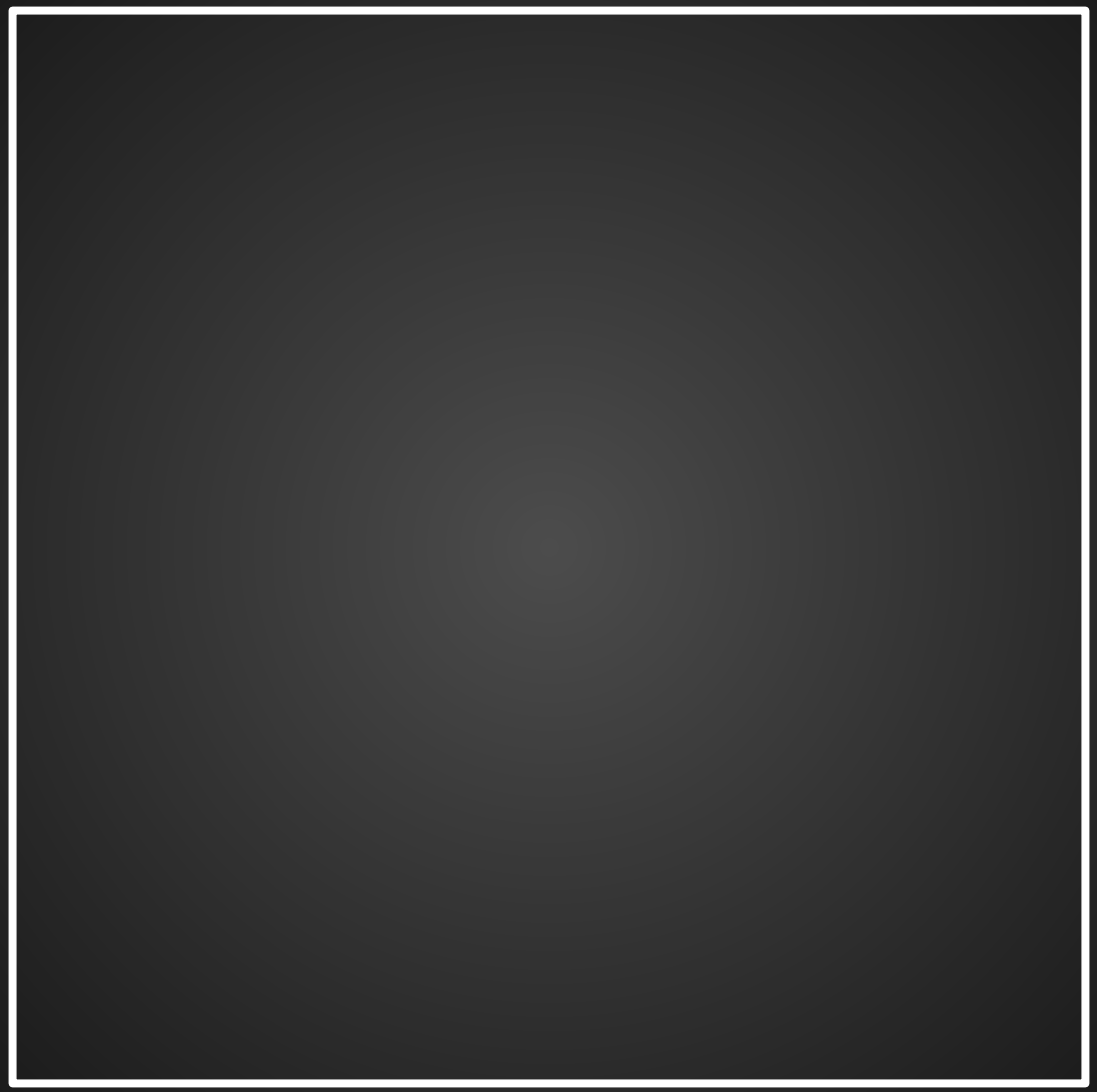


Separate responsibilities to preserve API stability.

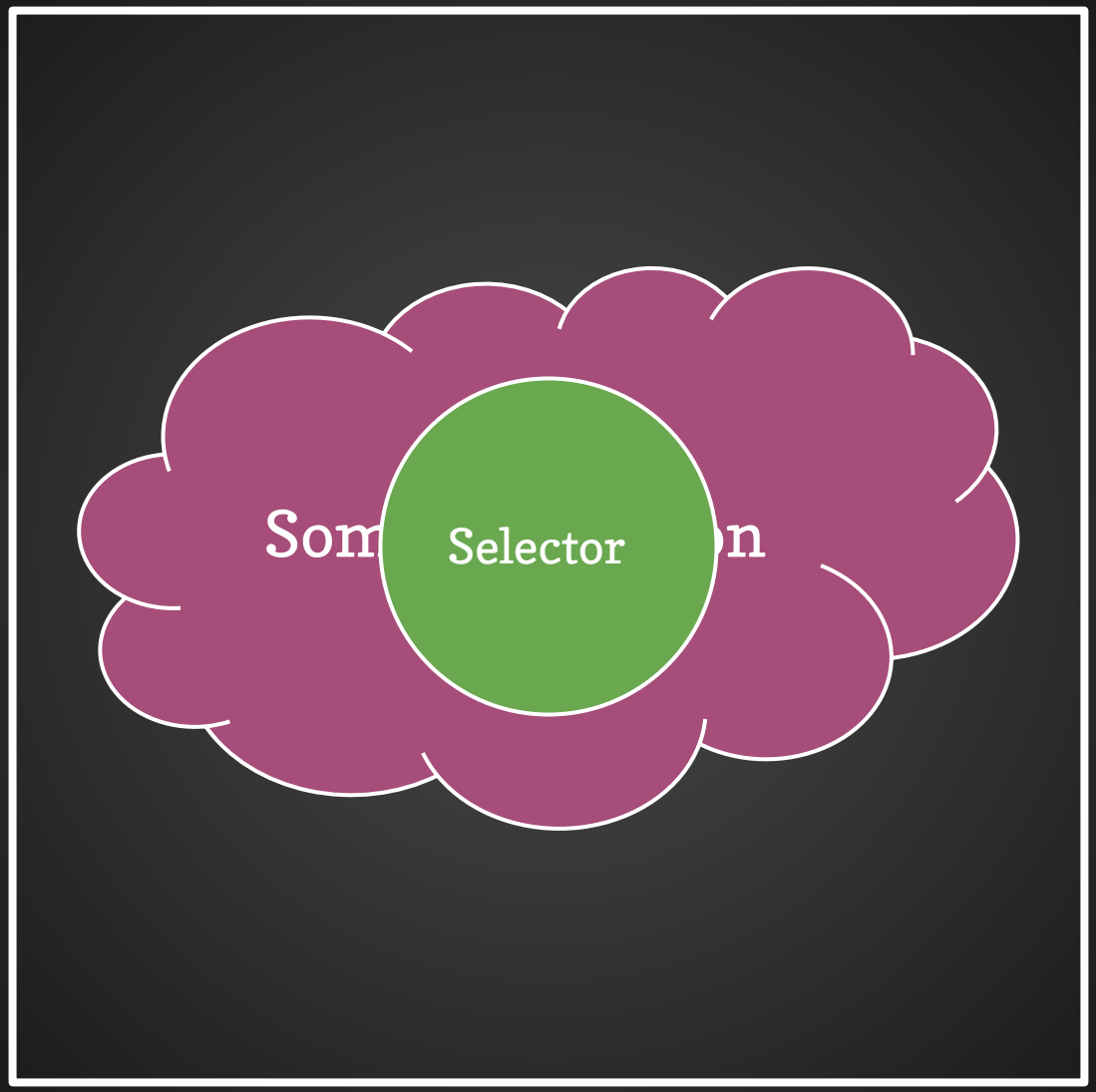
yt


API
IO, parallelism, indexing, processing,
iteration, filtering, ...

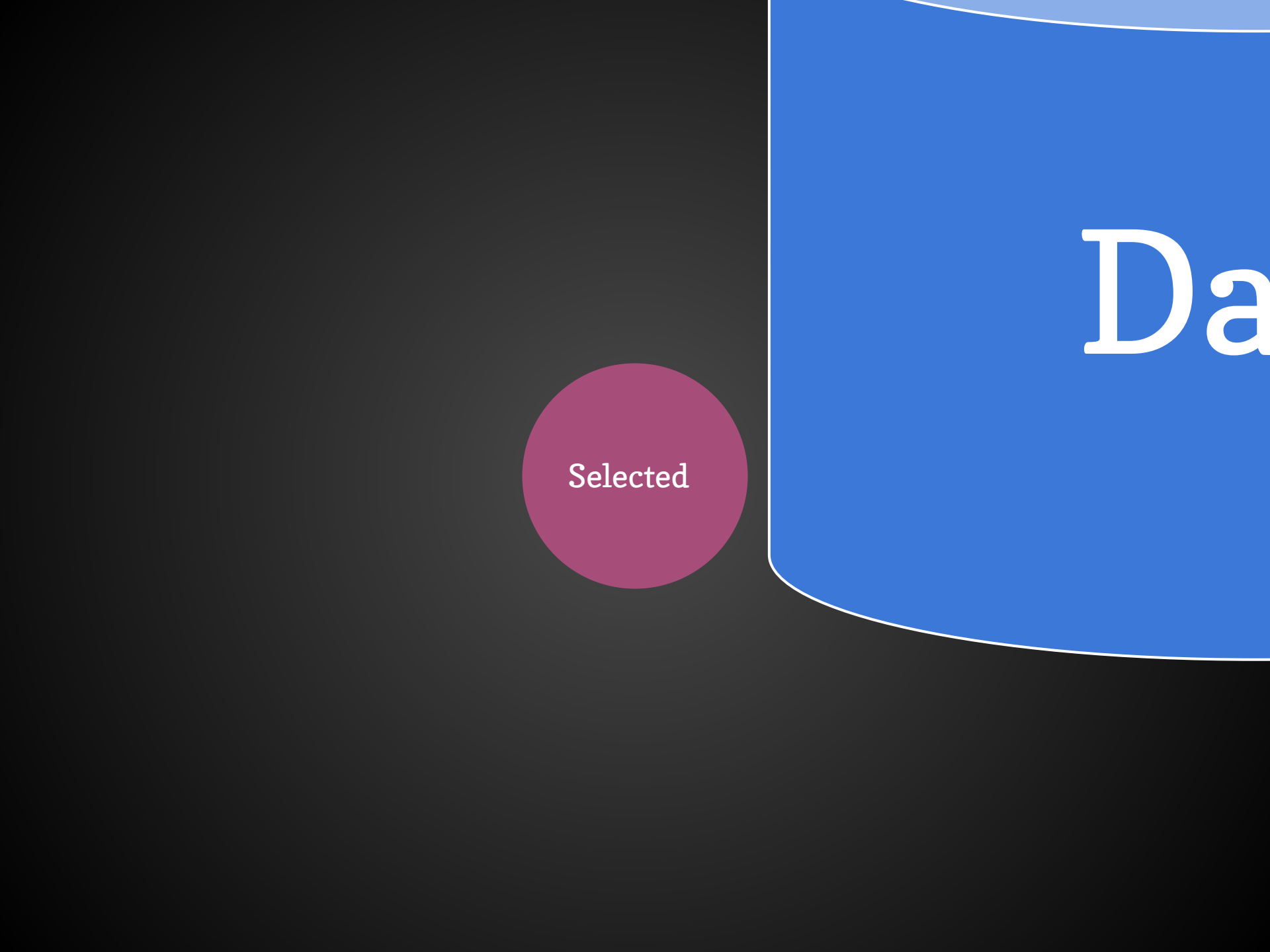
# Abstract data sources and indexing.

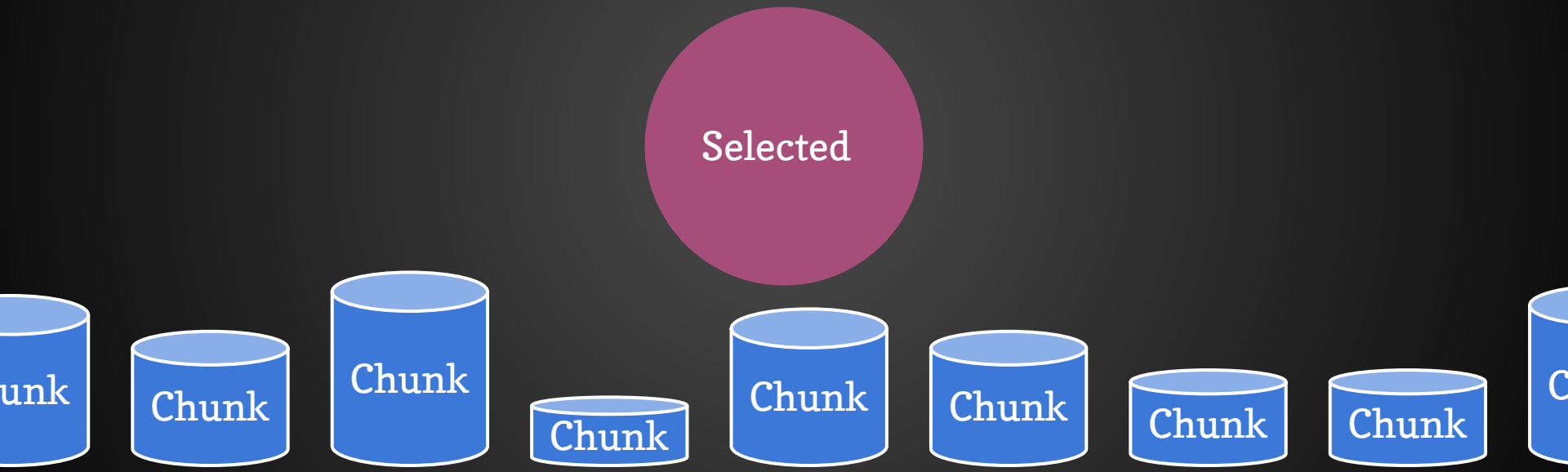



Some Simulation


Som
Selector
n

Selected


Da
Selected

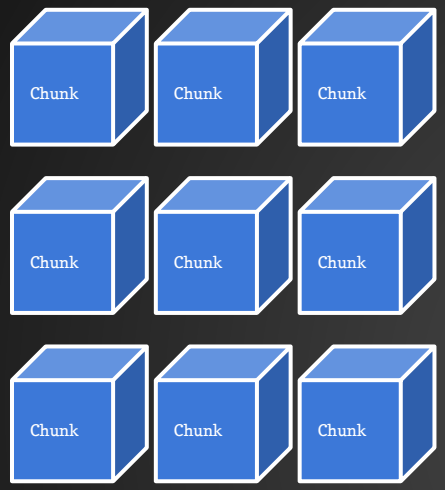

Selected
unk
Chunk
Chunk
Chunk
Chunk
Chunk
Chunk
Chunk
Chunk
C


Chunk


Values
Coordinates
Extent


Values
Coordinates
Extent
Components


Values
Coordinates
Extent
Components
Value Neighbors
Block Neighbors



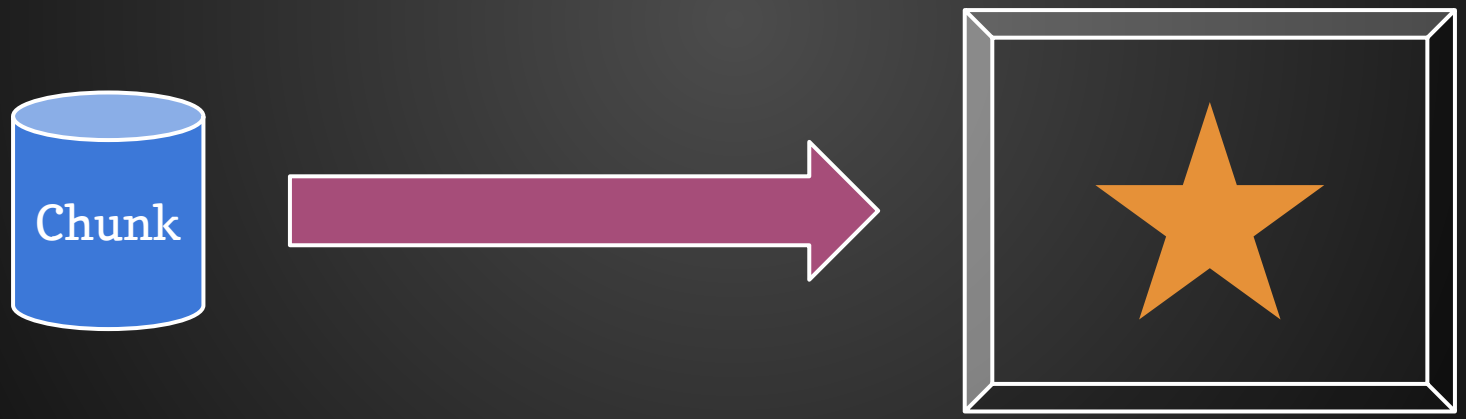

Values
Coordinates
Extent
Components
Value Neighbors
Block Neighbors

# High-Level Operations

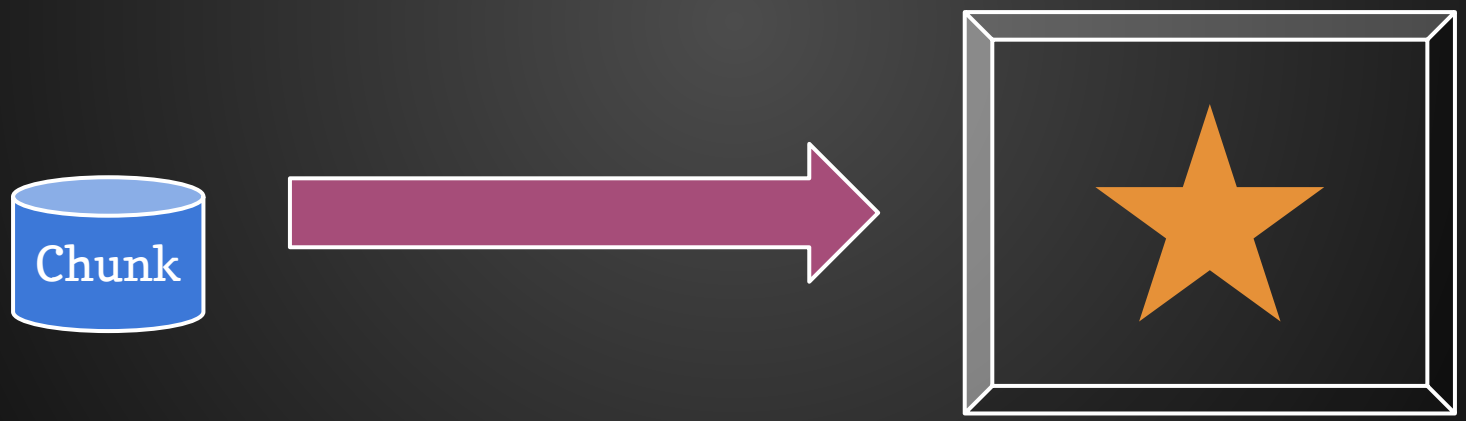
# High-Level Operations

# High-Level Operations

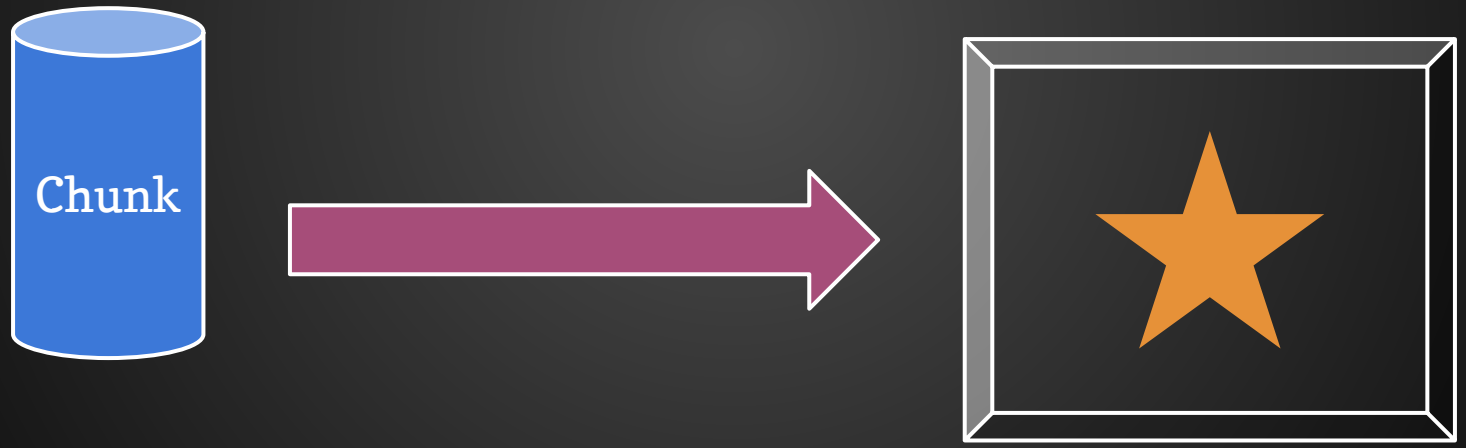
# High-Level Operations



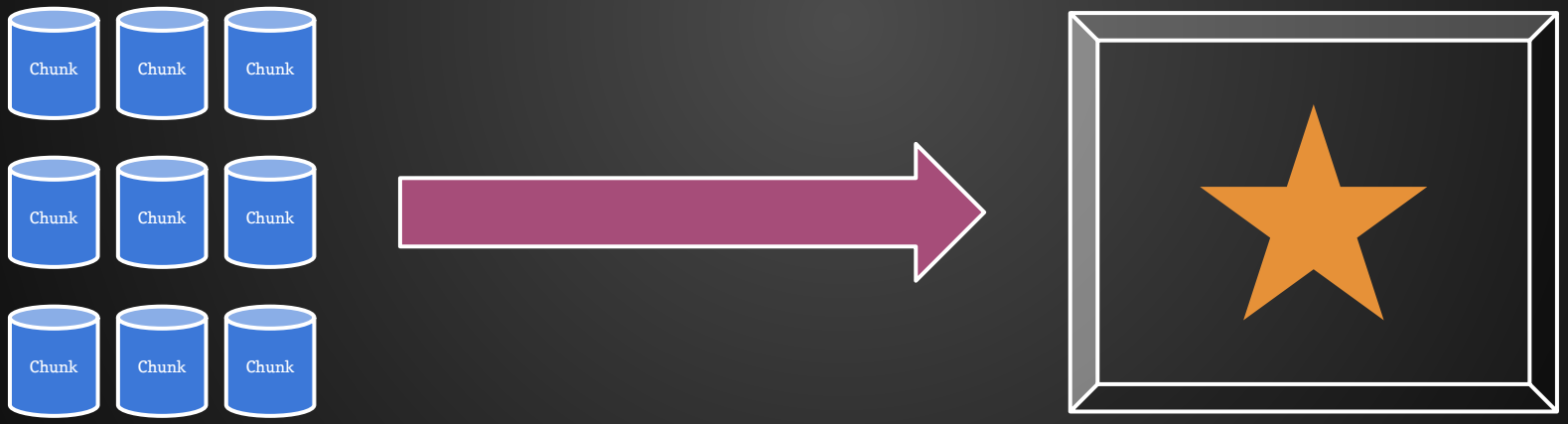
# High-Level Operations



# High-Level Operations



# High-Level Operations



# High-Level Operations

Process data.

# Low-Level Operations

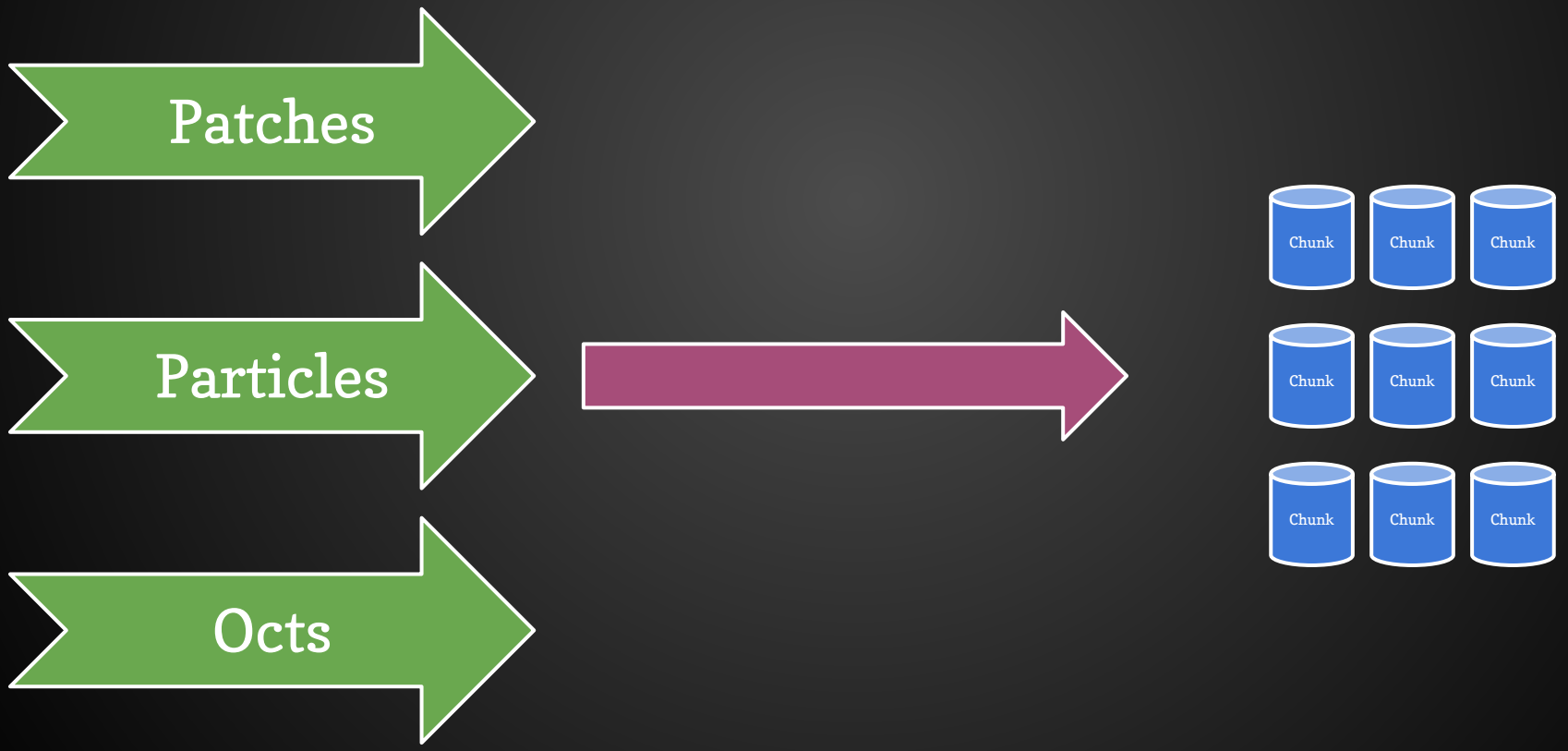
# Low-Level Operations

# Low-Level Operations


# Low-Level Operations

# Low-Level Operations

# Low-Level Operations



# Low-Level Operations

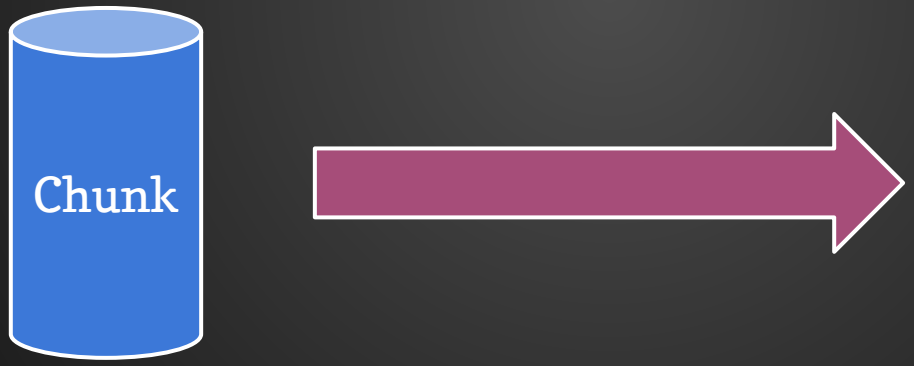
# Low-Level Operations

# Low-Level Operations

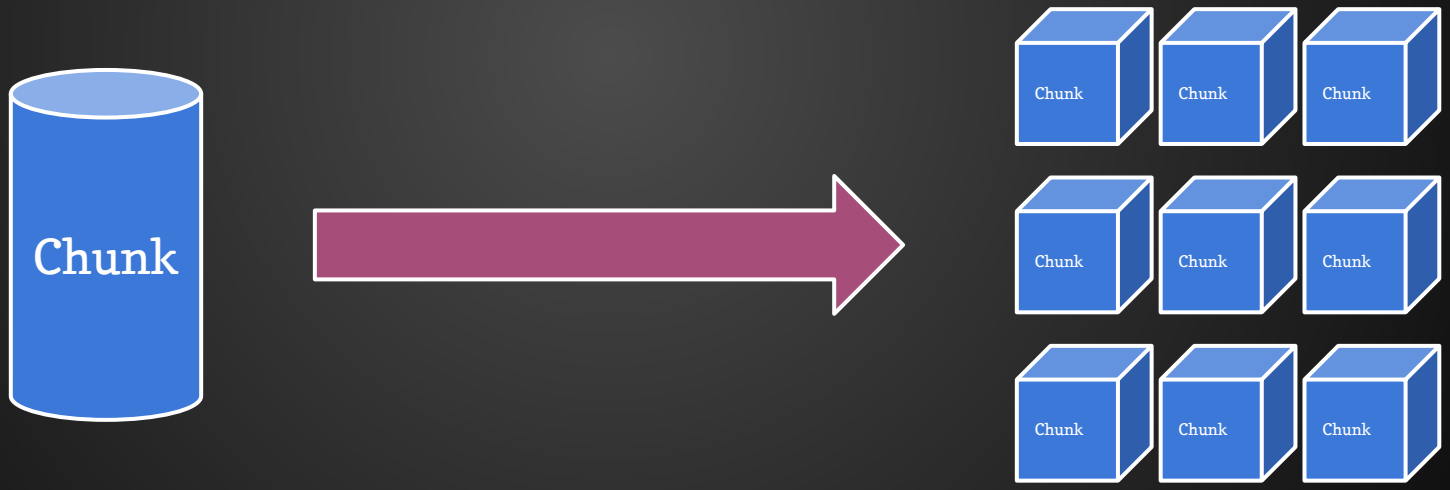
# Low-Level Operations

# Low-Level Operations

# Low-Level Operations



# Low-Level Operations



# Low-Level Operations

Index data.

# Software Infrastructure

# Software Infrastructure

# Software Infrastructure


Math

Solvers
Algorithms
IO
Parallelism
Analysis
Visualization

# Software Infrastructure


Math
Parallelism
Algorithms
Visualization

Solvers
IO
Analysis

# Software Infrastructure


Math
Parallelism
Algorithms
Visualization
IO

Solvers
Analysis

# Software Infrastructure


Domain specificity is a transient phenomena for infrastructure.

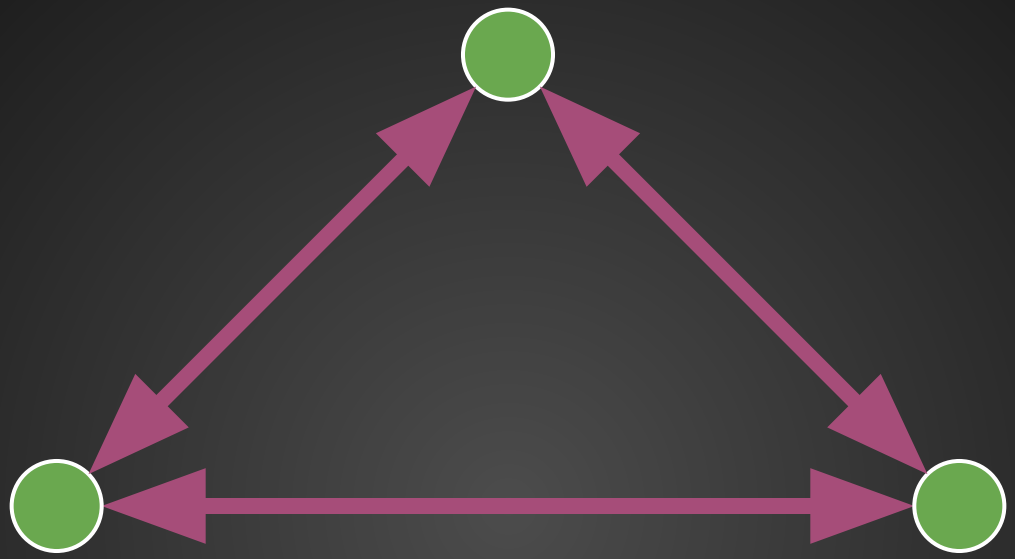
# Software Infrastructure

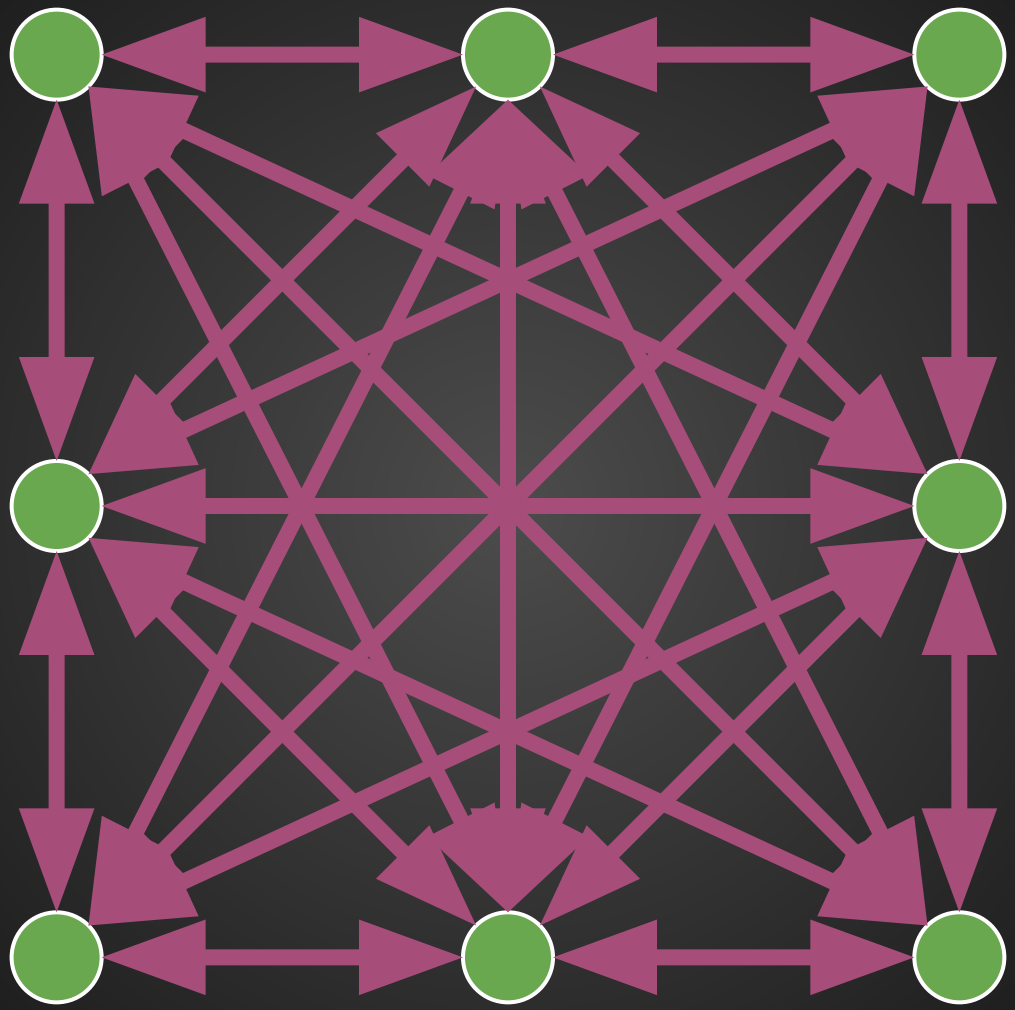

Domain specificity is a lasting phenomena for implementations.

# Not Invented Here

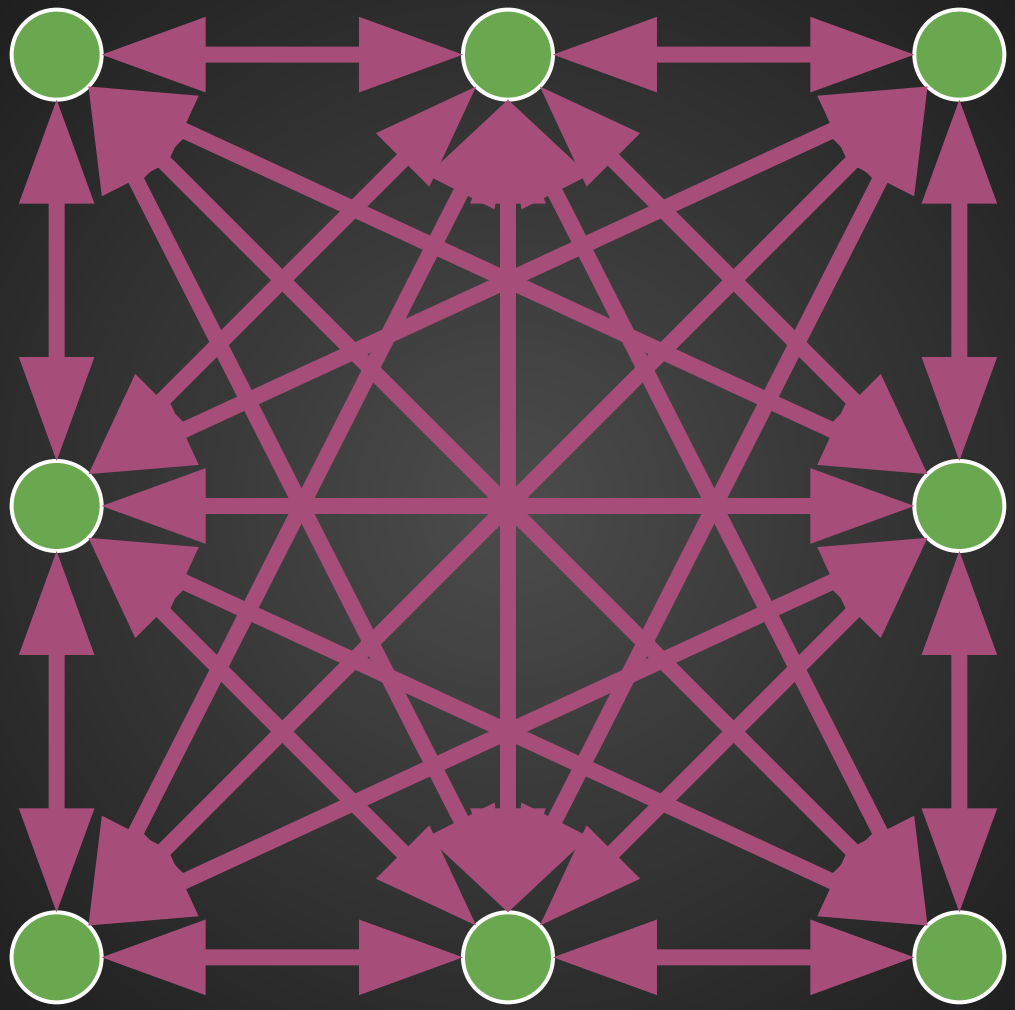
# Not Invented Here

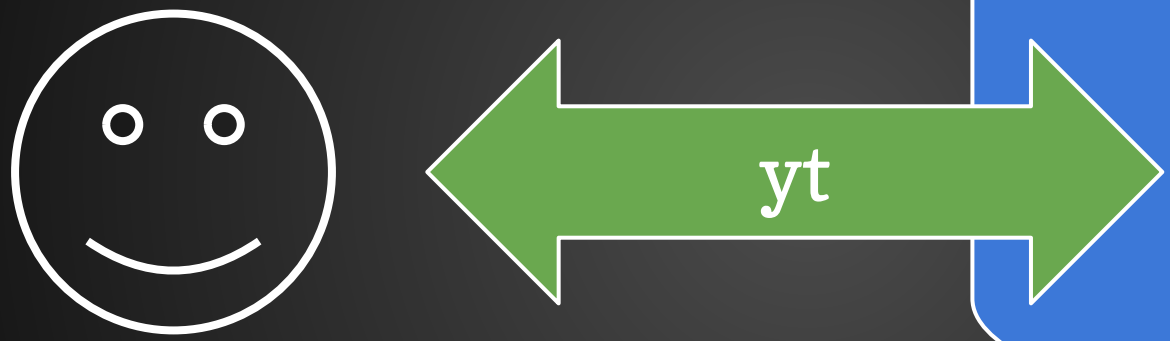
(a space-filling effect)

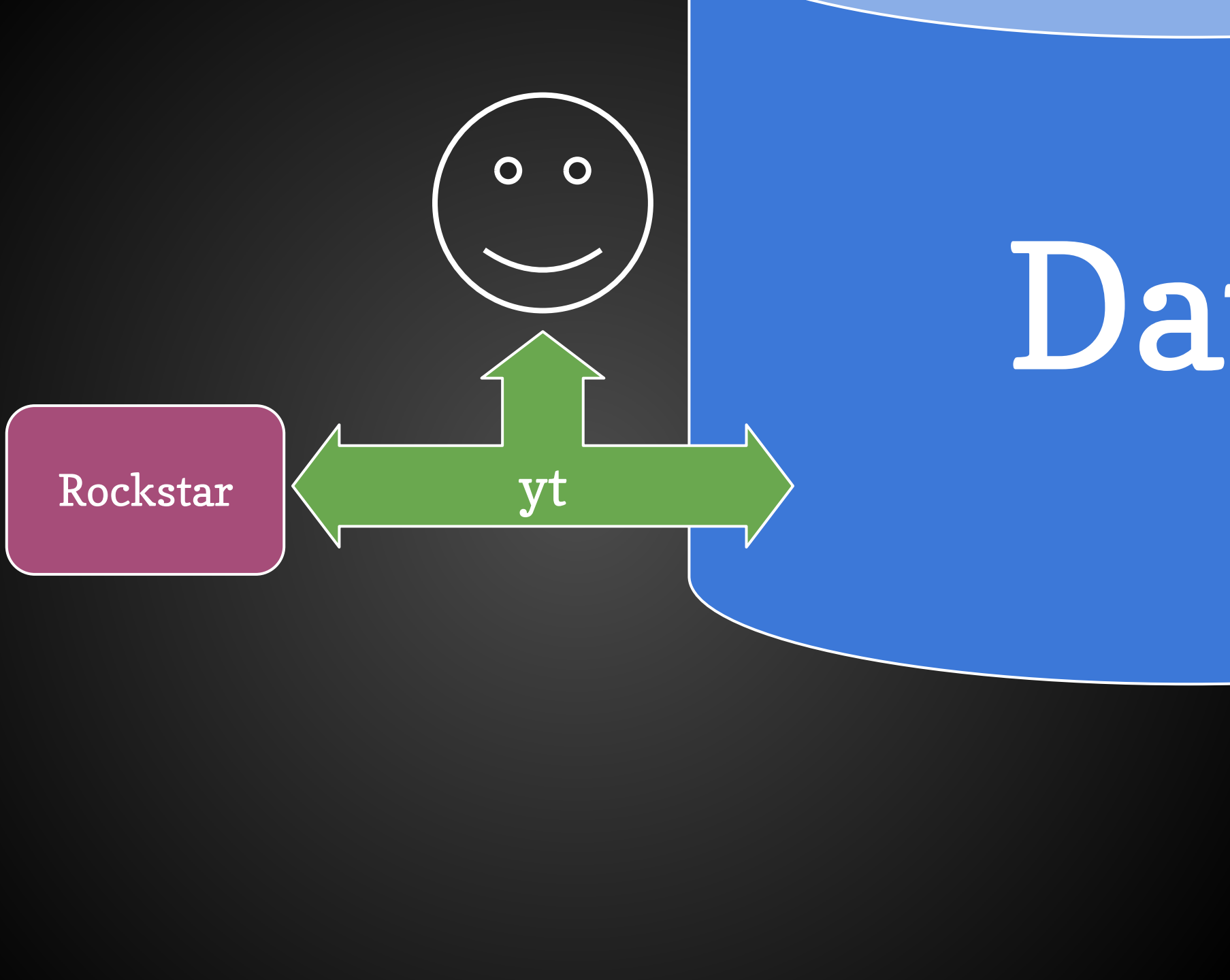


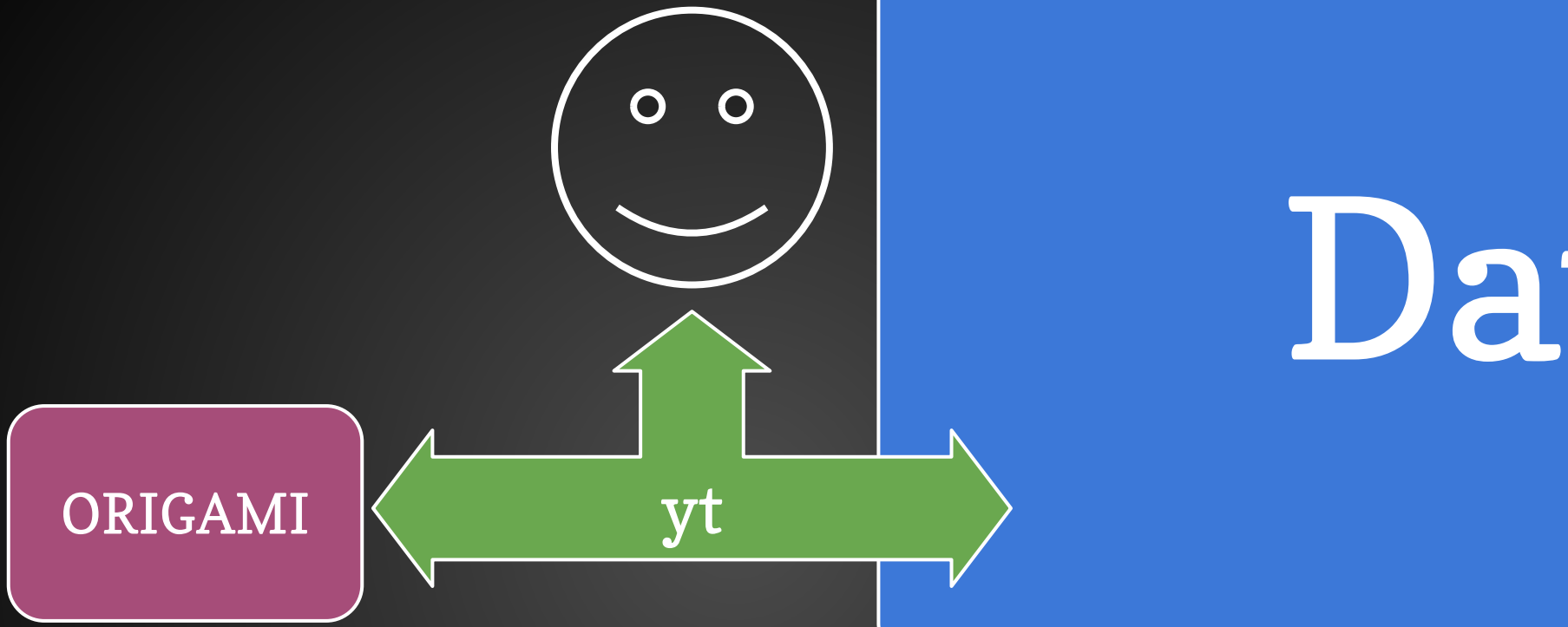


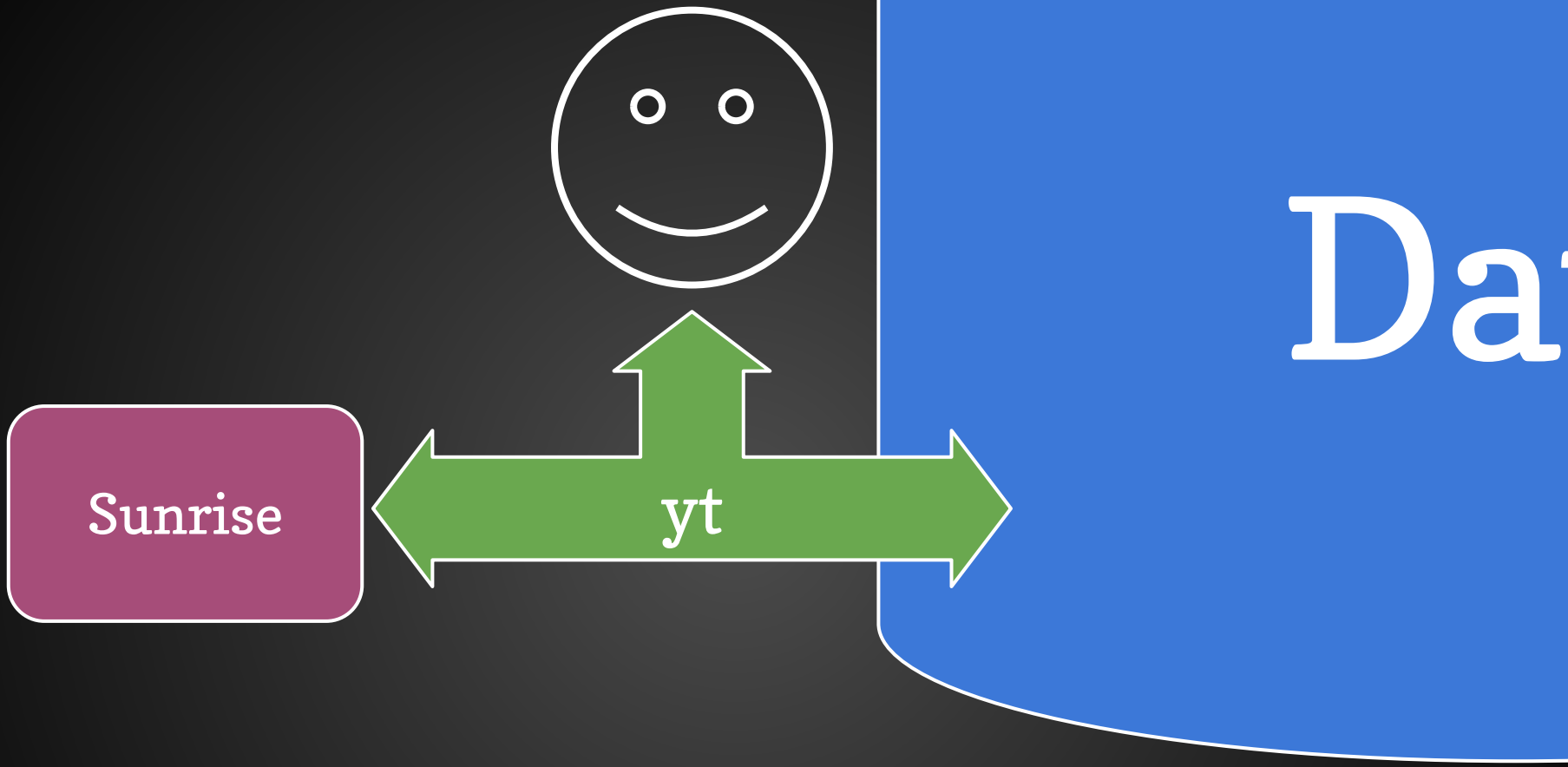
# Metcalfe's Law

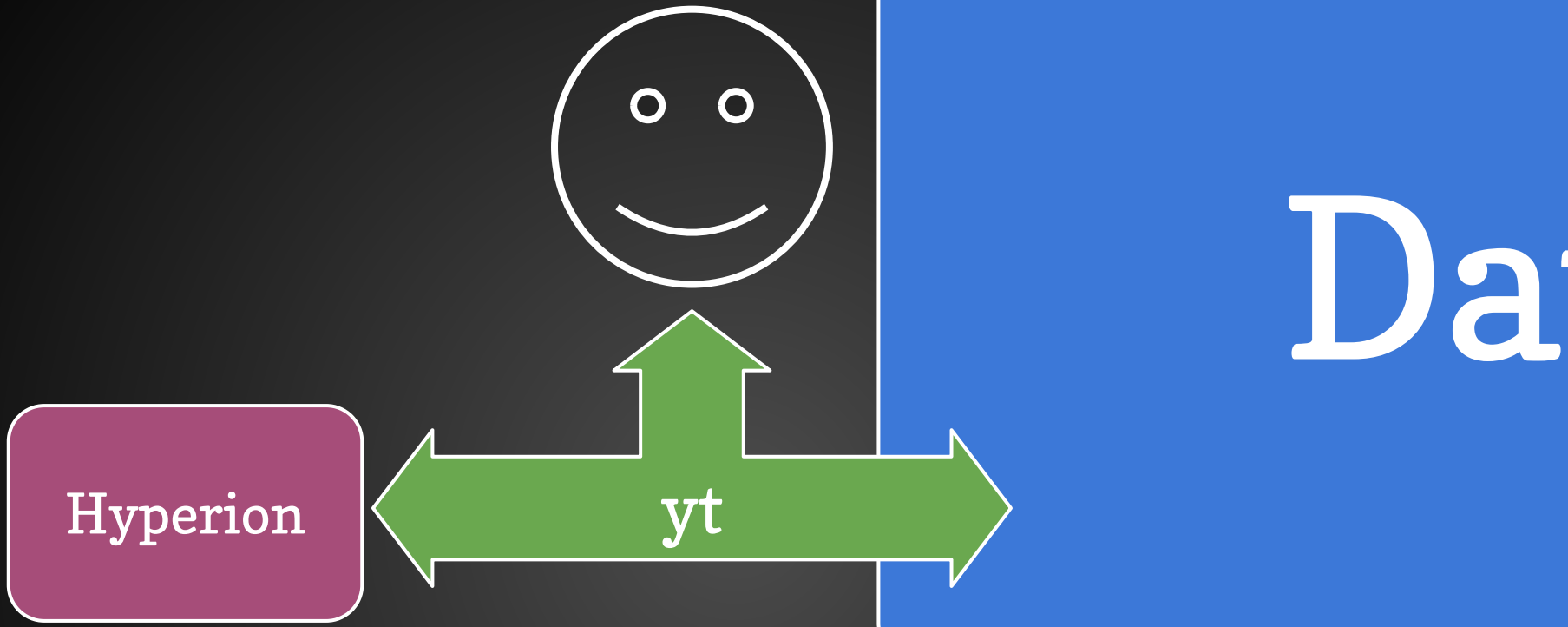


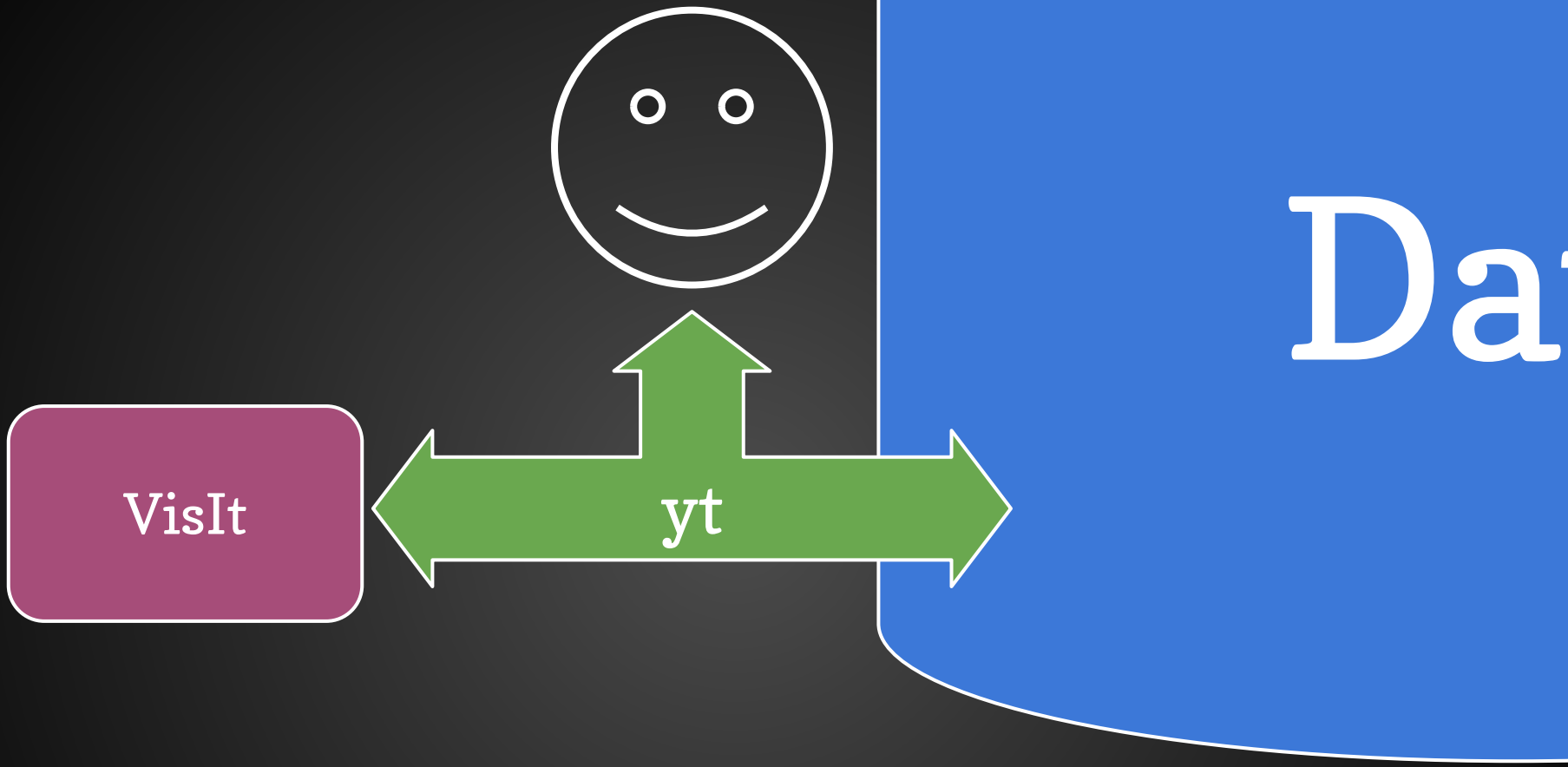

yt

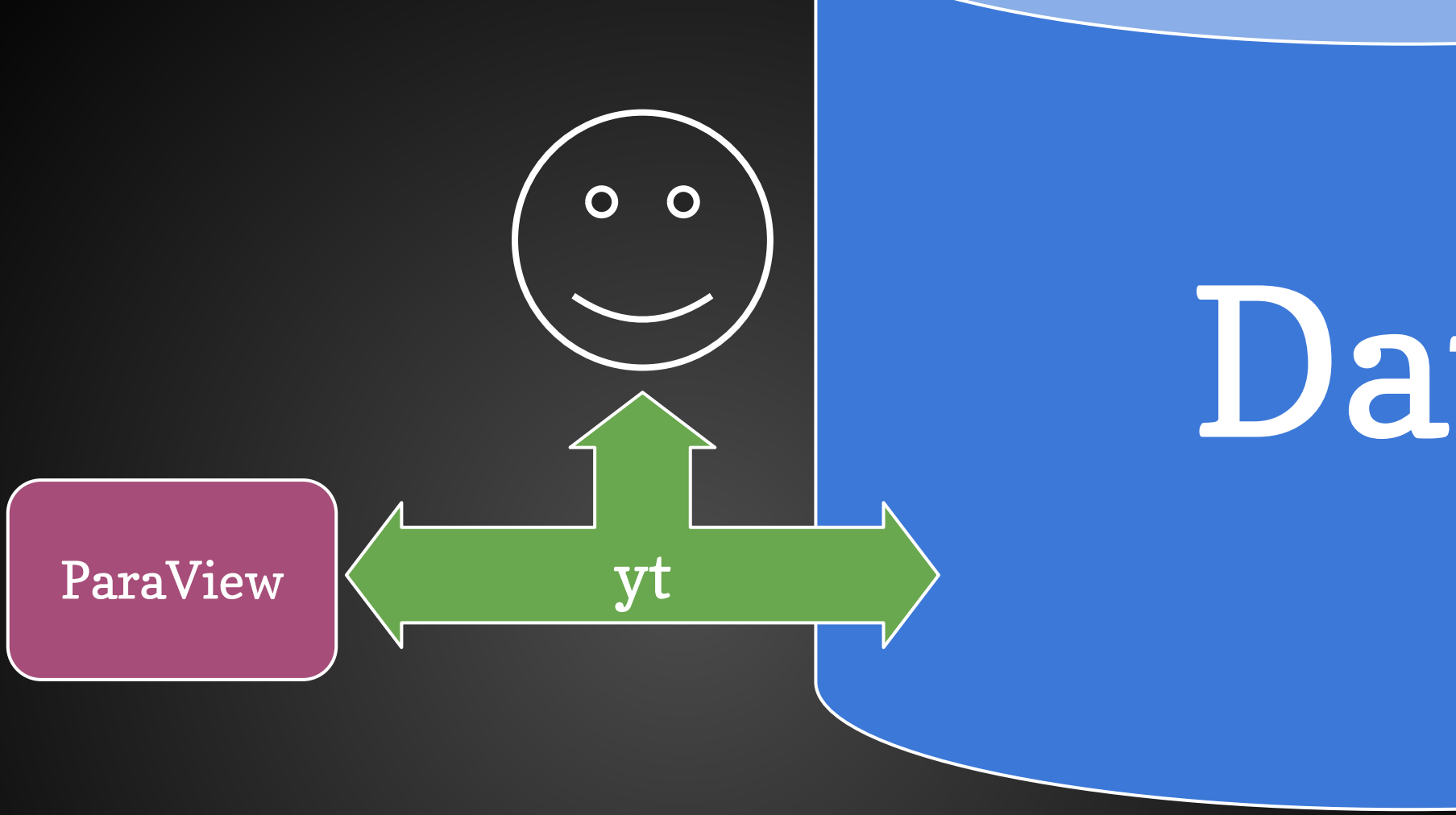

Da
Rockstar
yt

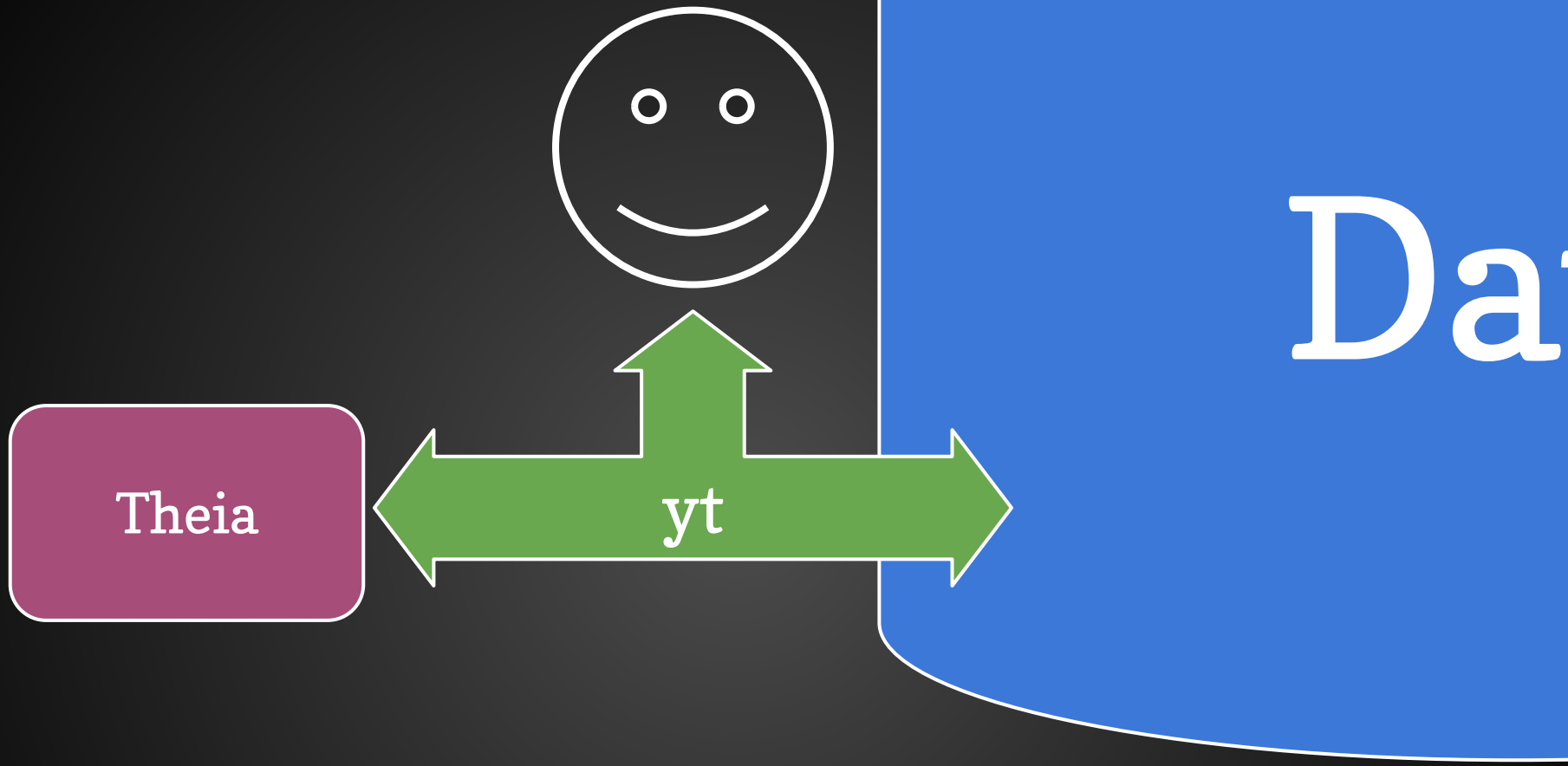

Da
ORIGAMI
yt


Da
Sunrise
yt

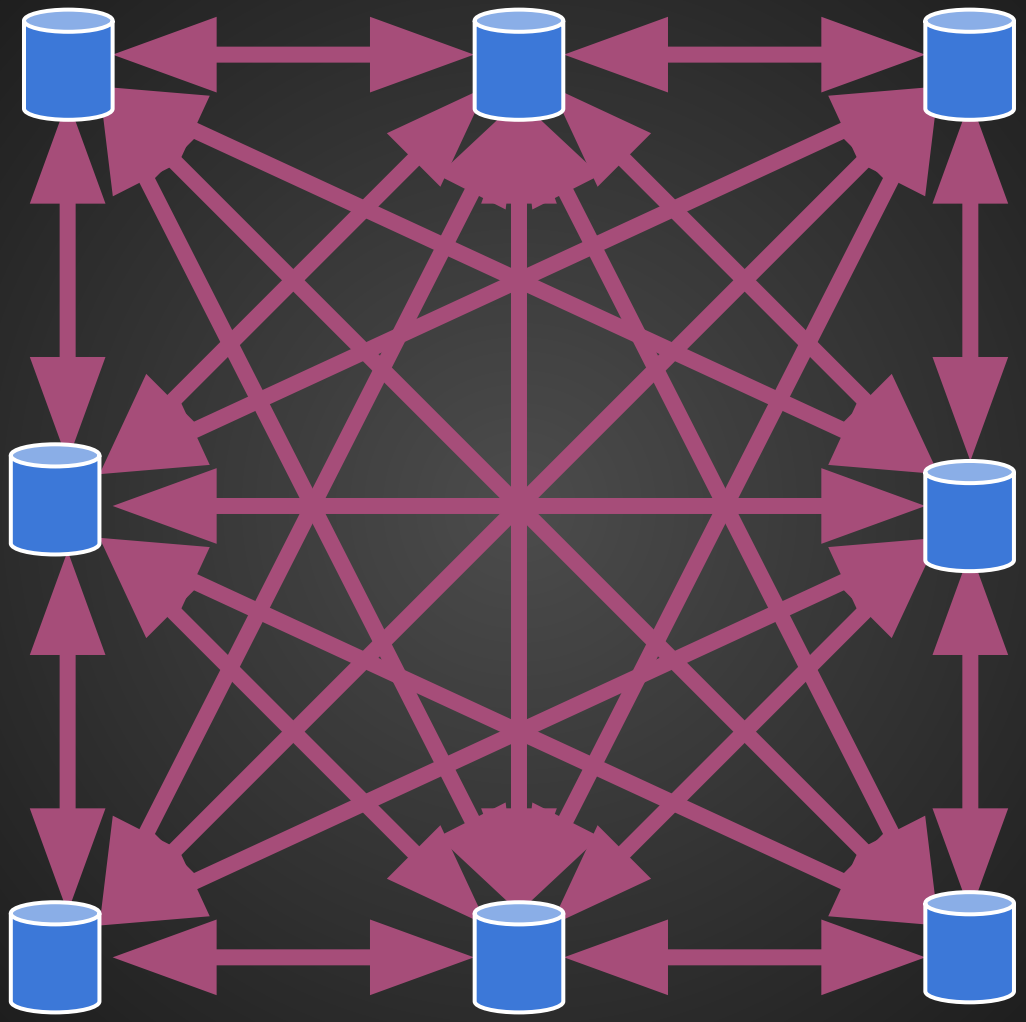

Hyperion
yt


Da
VisIt
yt


Da
ParaView
yt


Theia
yt

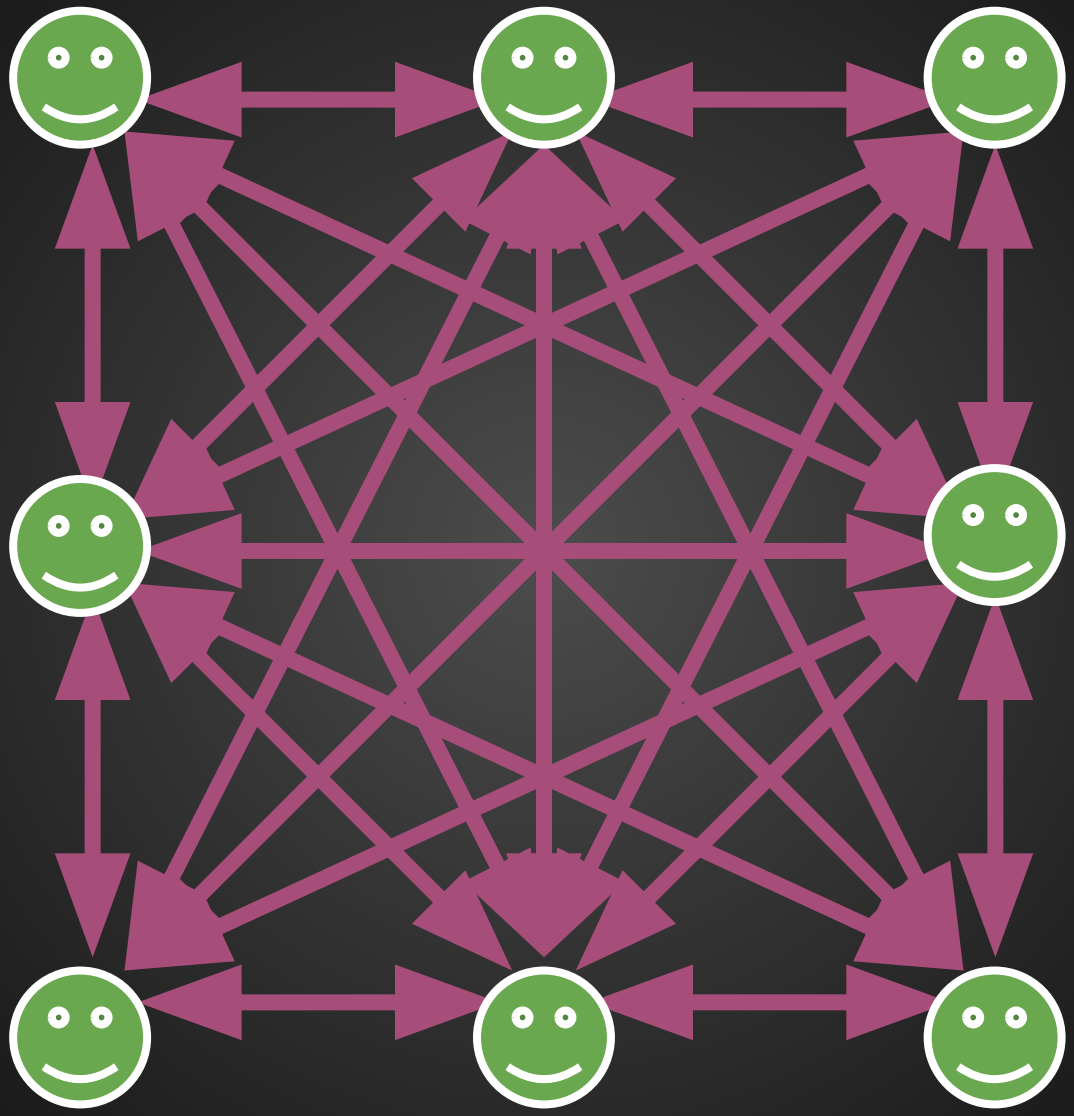
# Warren's Law



“Exascale is made of people!”

- Mike Warren, 2014

# People



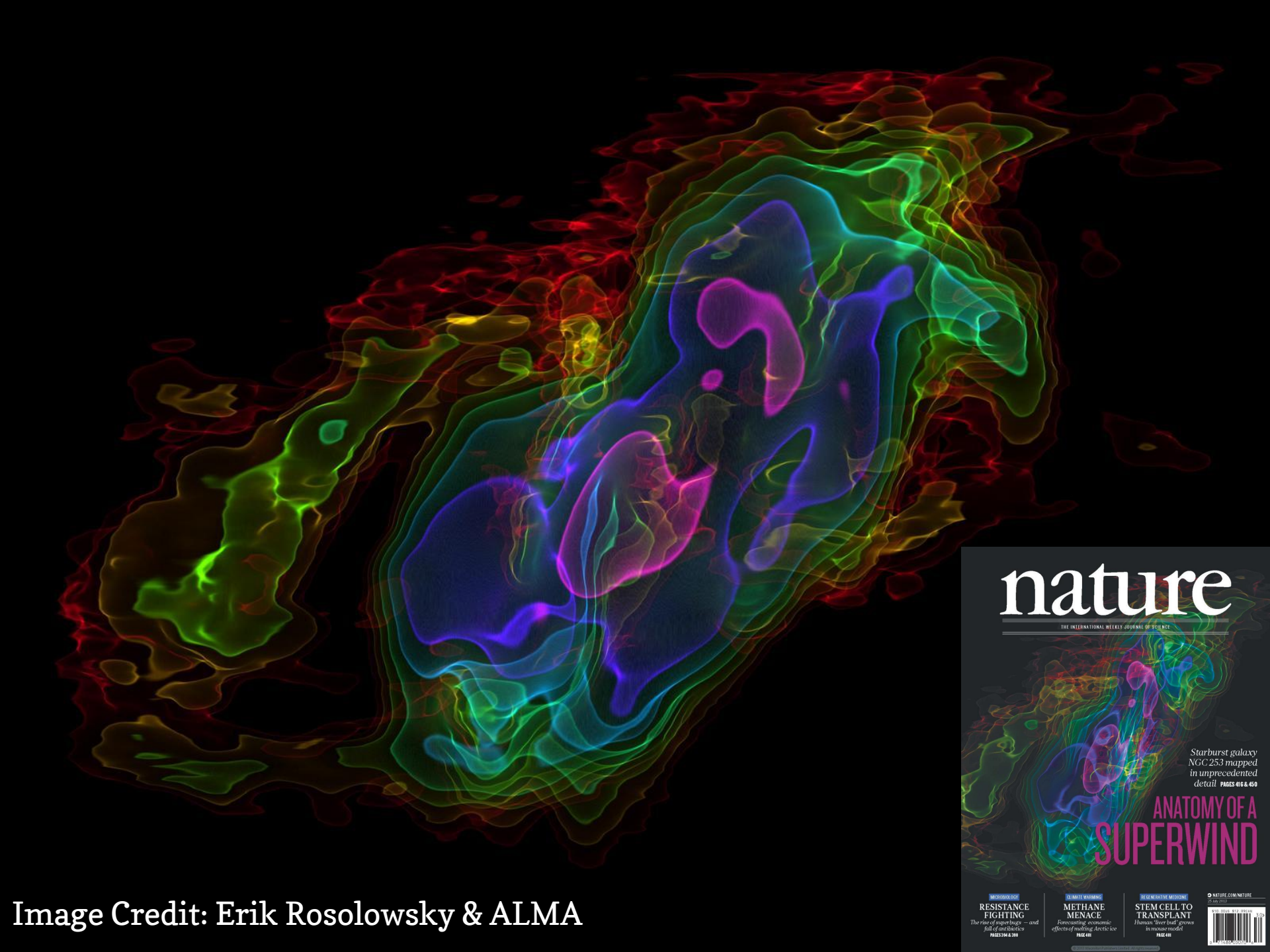
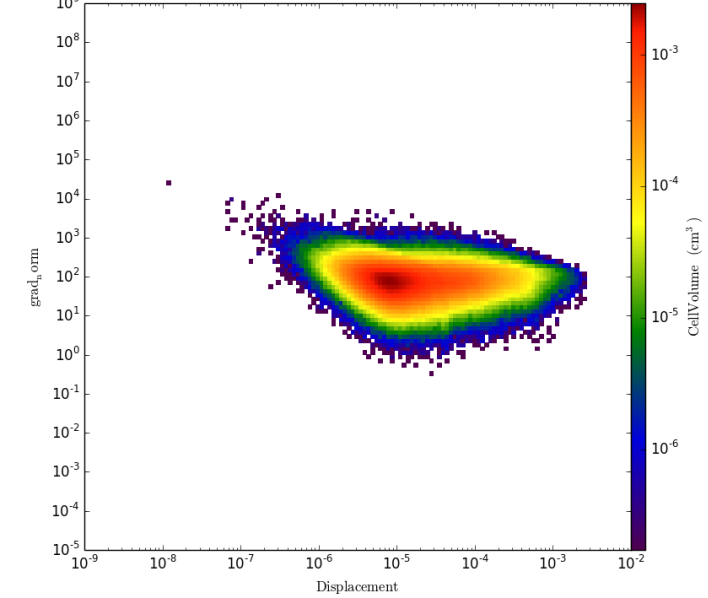
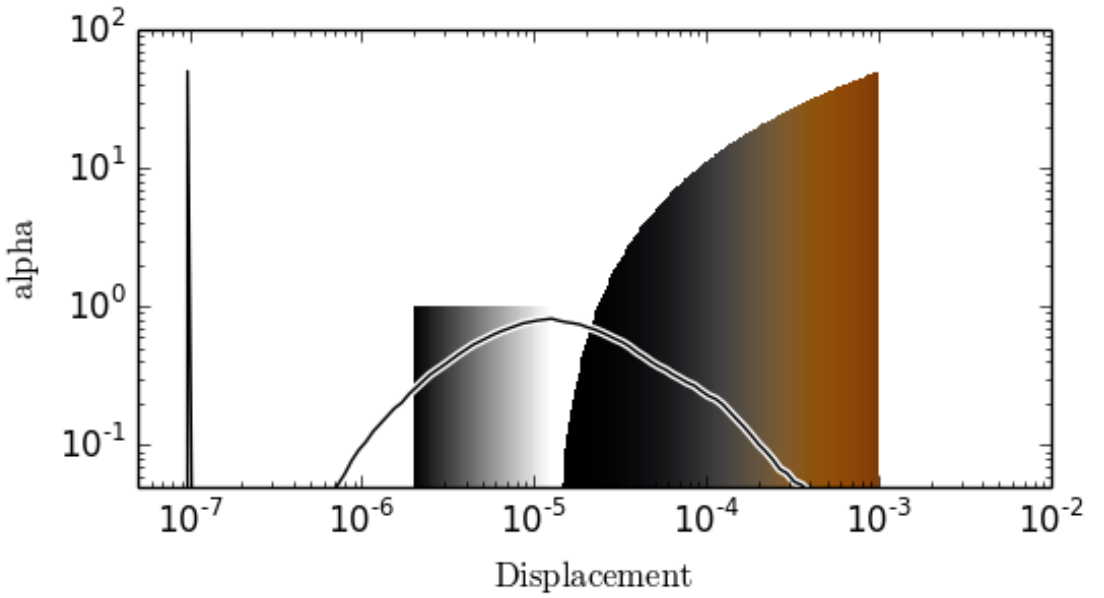
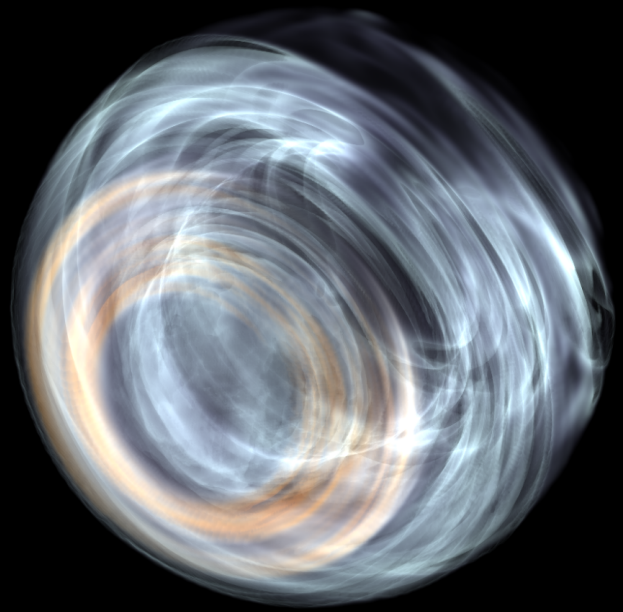


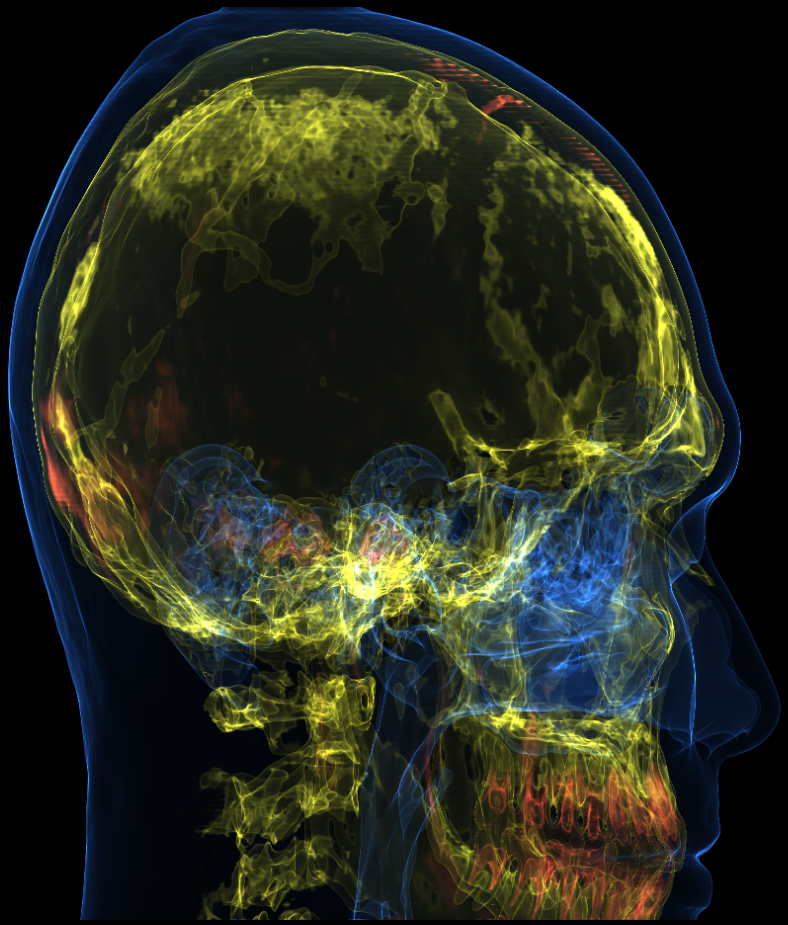

Image Credit: Erik Rosolowsky & ALMA



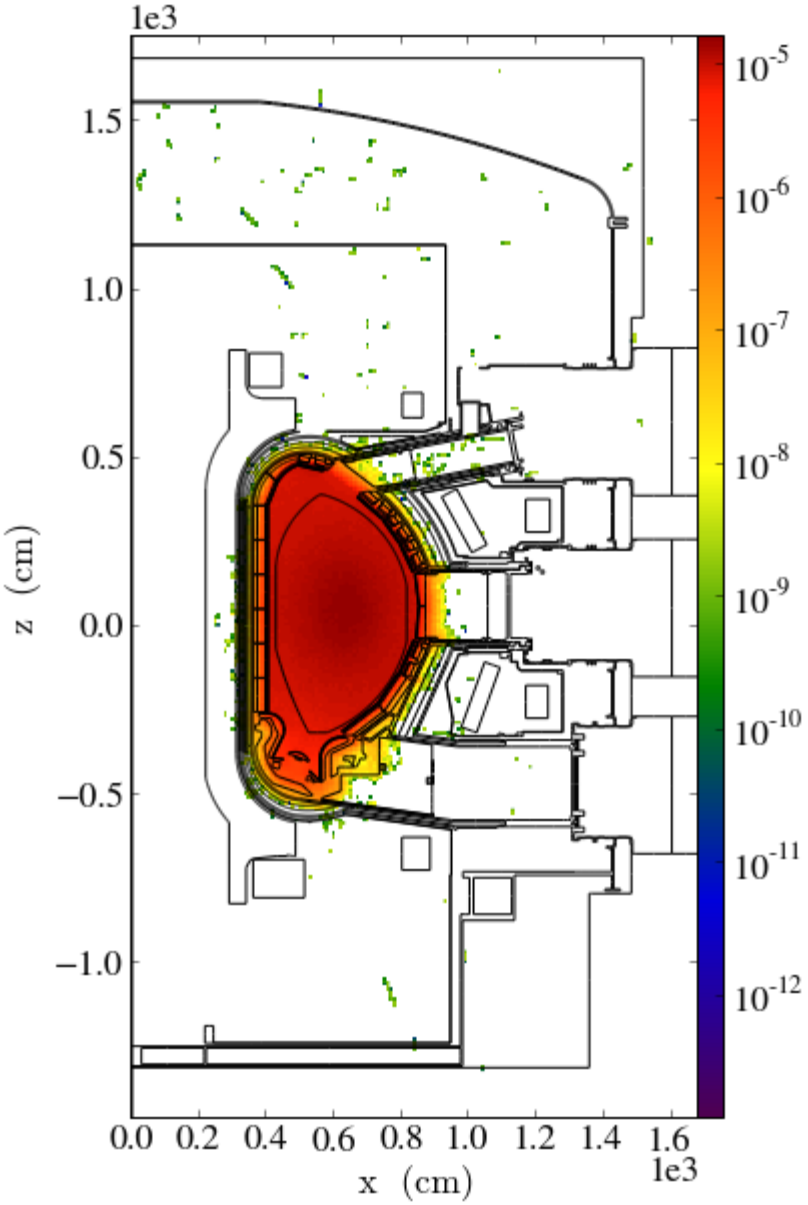

Holtzman et al

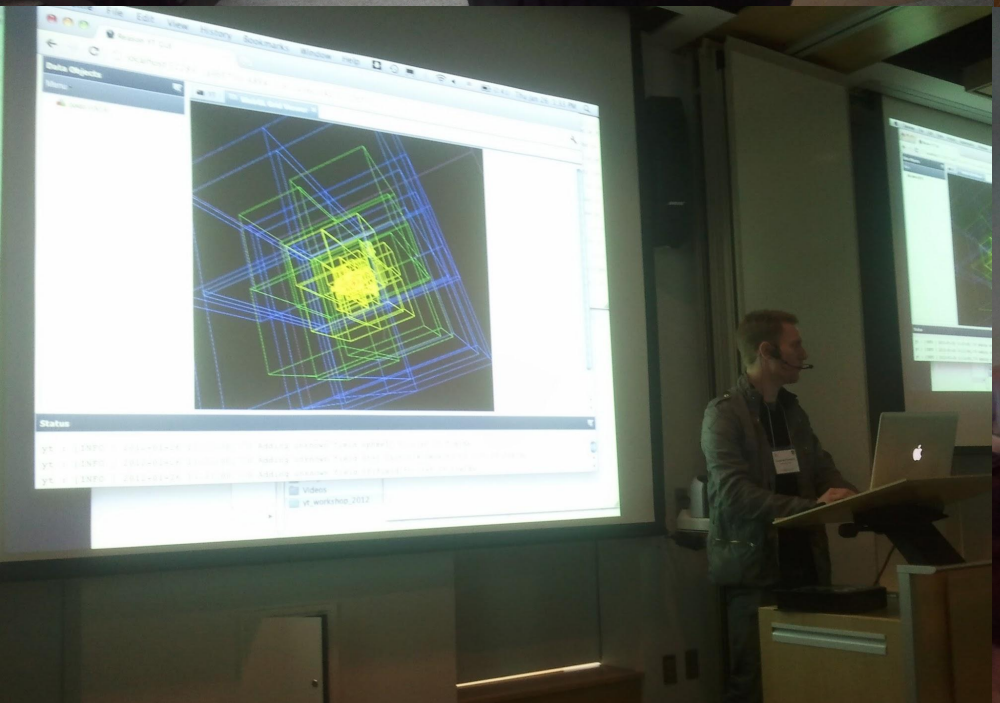




Neurodome and ITER

SubbaRao, Fisher, Shriwise





# Thank you.

Tom Abel
Kenza Arraki
David Collins
Brian Crosby
Andrew Cunningham
Hilary Egan
Nathan Goldbaum
William Grey
Markus Haider
Cameron Hummels
Christian Karch
Steffen Klemer
Kacper Kowalik
Mike Kuhlen
Eve Lee
Sam Leitner
Yuan Li
Chris Malone
Josh Moloney
Chris Moody
Andrew Myers
Jill Naiman
Kaylea Nelson
Jeff Oishi
Jean-Claude Passy
Mark Richardson
Thomas Robitaille
Anna Rosen
Doug Rudd
Anthony Scopatz
Devin Silvia
Sam Skillman
Stephen Skory
Britton Smith
Geoffrey So
Casey Stark
Elizabeth Tasker
Stephanie Tonnesen
Sebastian Trujillo-Gomez
Matthew Turk
Rick Wagner
Andrew Wetzel
John Wise
John ZuHone