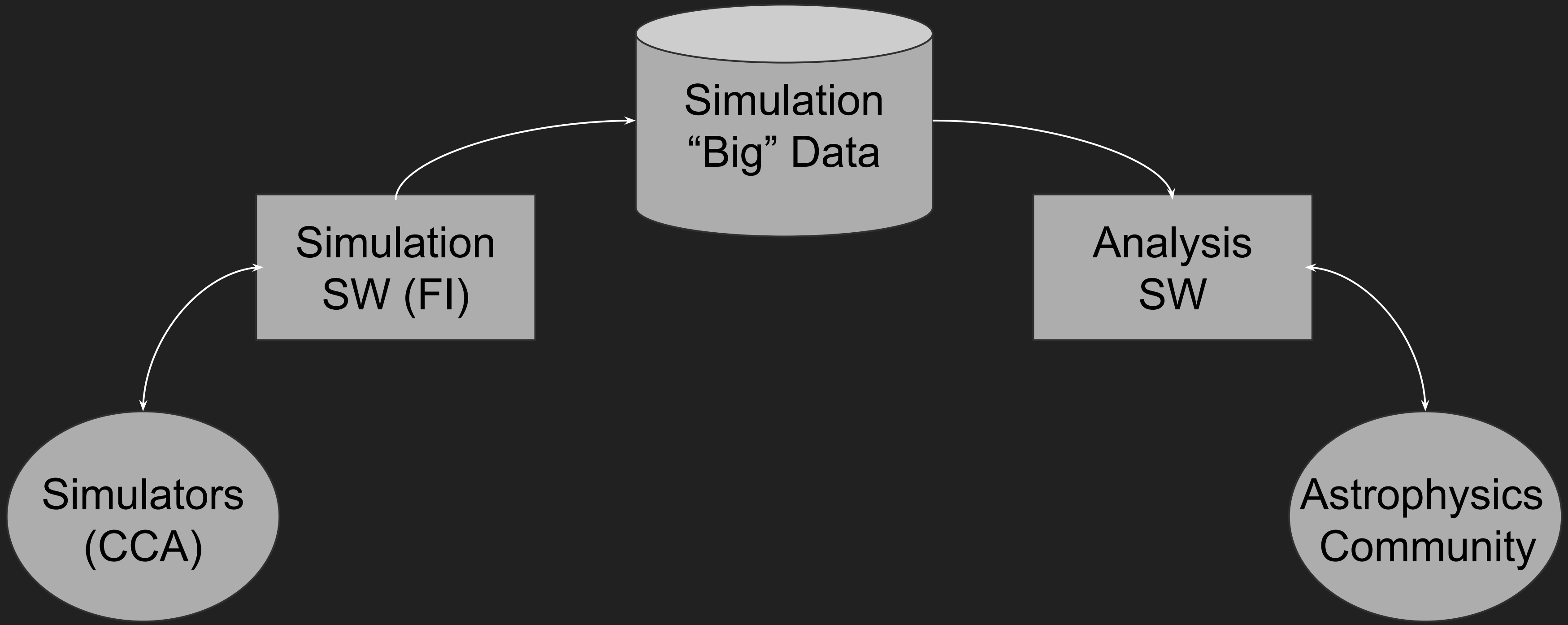
Flatware Services: CCA Simulation Repository

Sharing Astrophysical Simulation Data

Dylan Simon, Austen Gabrielpillai
Shy Genel, Chris Hayward, rachel somerville

The Problem



Simulation “Big” Data

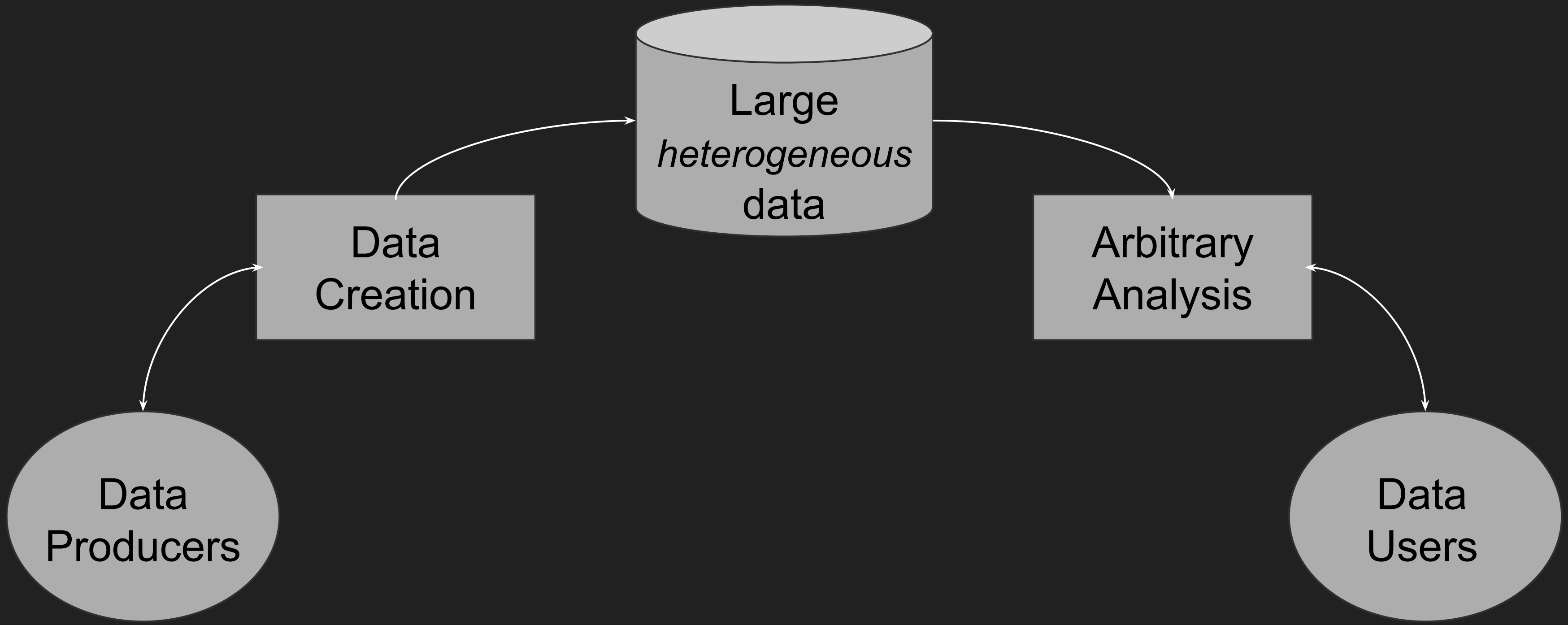
Simulation SW (FI)

Analysis SW

Simulators (CCA)

Astrophysics Community

The Problem



Large *heterogeneous* data

Data Creation

Arbitrary Analysis

Data Producers

Data Users

Approaches

- Download big files: bring data to users
  - Easy to provide, allows full, arbitrary access
  - Significant user investment: need enough storage/compute, understand data formats
- Provide computational environment: bring users to data
  - Data producer investment: provide computational resources, interface
  - FI: unrestricted HPC/shell access to...
  - SciServer: complete walled-garden ecosystem
  - Requires balancing user scaling and restrictions
- Provide limited querying capability, partial downloads
  - Query interface defines abilities, development vs. computational cost
  - SQL: users write SQL text query, execute with some time/size limit
  - Illustris: API (and python client)
  - CosmoHub: SQL+data exploration, limited API

Challenges

- Documentation
    - Users need to understand how to find what they want, how to use what they get
- Generalizability
    - Custom-made solutions specific each data set
    - Hard to add new datasets, adapt analysis code to new data
- Performance
    - Big transfers and big computations have cost
    - Users (and providers) can wait days for results

Astrosims Data Repository

- Goal: bring relevant subset of data to user
  - Intuitive, well-documented: direct, consistent access to data, structure
  - General, extensible: easy to add & query new datasets without changing code
  - Fast, performant: filter, explore, download data interactively, low-cost
  - API: allow direct, seamless access from code

- Focus on tabular (catalog) data
  - Flexible, dynamic schema (field definitions)
  - Build on elasticsearch (lucene) database
- Demo

Future work

- Standardize field definitions, units
- Access to snapshot volume data
    - Formats are fairly specific to datasets
    - Standard formats (yt)? Conversions can be lossy
    - Standard API/links based on catalog objects, dataset-specific operations
- More complex queries
    - Multiple fields (“*x* > *y*”)
    - Joins/cross-matches between datasets
- Meta-structure, discovery across datasets
- Improve UX design, style
- Generalizations, other applications
- Hosted at SDSC on kubernetes as PoC, built on docker hub
    - Now available for more public data serving applications