From GPUs to IOUs: Tales of Computing and Cost Recovery in the Core

Sean Taylor

Former Director of Research Computing and Bioinformatics, Seattle Children's Research Institute

Background

- Biology has become a data science
- NIH emphasizes and rewards having access to integrated genomics
  - Drives high demand for quantitative genomics skills and access to scalable computing infrastructure
- Research Scientific Computing
  - Access to a team of highly trained computational biologists, data scientists, data engineers, and software developers
  - Access to a fully integrated and supported computing environment, built for and maintained by scientists who understand the work you do

What HPC's can do

And where they will let you down

Establishing Scientific Computing at SCRI

Prior to 2015

- No compute infrastructure for bioinformatics
- A few labs engaged in computational approaches
- No general bioinformatics support
- Limited literacy and computational expertise

2016

- Linux approved
- Autobot linux servers deployed
- Launch training courses in computing and bioinformatics

2017-2019

- Cybertron HPC deployed
- Integration of research storage services
- Support for high-throughput analytics
- Expansion of course curriculum
- Growing number of labs engaged in computational approaches
- SCRI begins recruiting faculty using HPC

2020-2023

- HPC Core business established, with automated billing and reporting processes
- Full support for common development IDEs
- Centralized Bioinformatics support team (and growing)
- Supporting robust grant portfolio across all centers
- Robust community with growing literacy and computational expertise
- SCRI actively recruiting faculty with intensive computational needs

2024-2025

- Sasquatch HPC deployed
- Posit for integrated IDE support
- Genomics and Spatial CoLabs established
- Growing interest in AI/ML

Growth of SCRI Scientific Computing

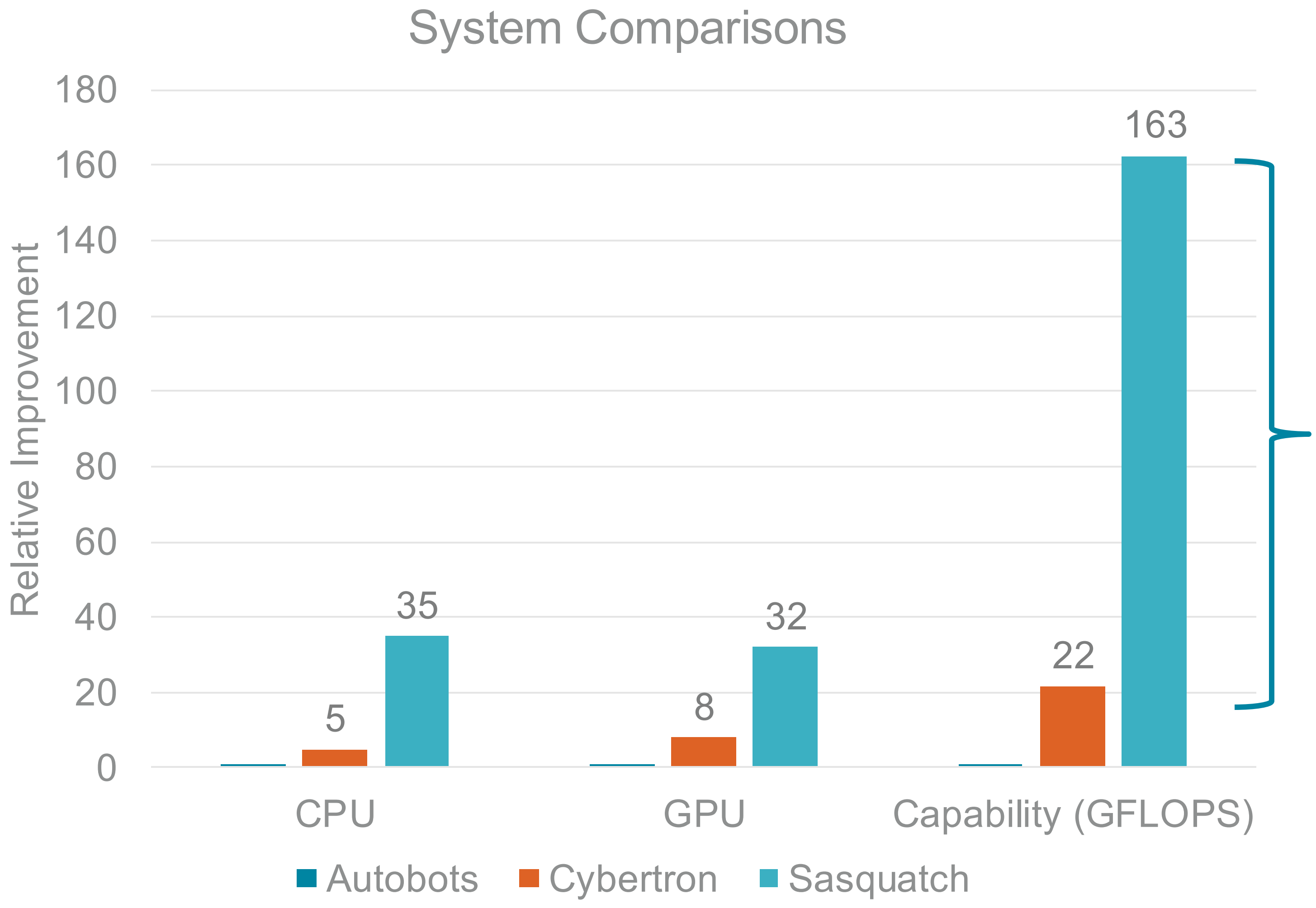
2015-2020

2017-2024

2024-

	Autobots	Cybertron	Sasquatch
CPU	176	872	6144
GPU	0	8 P100	32 A100
Main memory	768 GB	19.2 TB	57.3 TB
GPU memory	0	128 GB	2.56 TB
Capability	3,392 GFLOPS	73,568 GFLOPS	551,245 GFLOPS

Enhanced capability of Sasquatch



Sasquatch is >8x more powerful than Cybertron

Where Sasquatch shines

- Improved core density of 128 cores per node allows for more efficient multi-threading of jobs, greater parallel processing capability and improved resource utilization within a single compute unit

- HPS Parallel File system enabling concurrent data access from multiple compute nodes, significantly boosting I/O throughput and reducing data access bottlenecks for parallel applications.

- Deep learning models like AlphaFold for protein structure prediction benefit immensely from the A100's Tensor Cores, allowing for faster training and inference of complex models

HPC solves everything!...right?

Suitable workloads for HPC

- High degree of parallelism
- Large data volumes
- Computationally intensive tasks
- Highly scalable
- Open-source, programmatic execution
- Open-source Linux-native applications

- Some parallelism mixed with sequential steps
- Medium data volumes
- Moderately intensive tasks
- Limited scalability
- Pipeline structures
- Interactive, notebook execution
- Licensed, Linux-native applications

- Sequential workflow
- Small data volumes
- Computationally inexpensive
- Frequent interactive steps
- GUI interfaces
- Windows- or Mac-native applications

Suitable workloads for HPC: Examples

- WGS and WES Analysis
- De Novo Genome Assembly
- Single cell omics
- Large-Scale Molecular Dynamics (MD) Simulations
- Phylogenomics
- Deep Learning for Large Biological Datasets
- Imaging Analysis using ImageJ, CellProfiler, Napari, scikit-image

- RNA-seq Analysis
- ATAC-seq Analysis
- Metagenomics Analysis
- QTL
- GSEA

- Basic sequence alignment
- Manual curation of data
- Simple statistical tests in R or python
- Visualization of small to medium-sized datasets
- Microscopy applications such as ZEISS ZEN, Imaris, Arivis Vision4D

Upgrading people > upgrading tech

- The human factor is a bigger limitation than the technology. Our current technology stack is severely under-utilized.

- Most workflows require deeper configuration or retooling to take full advantage of HPC resources such as GPUs and MPI

- Most Sasquatch users are not optimally structuring workloads

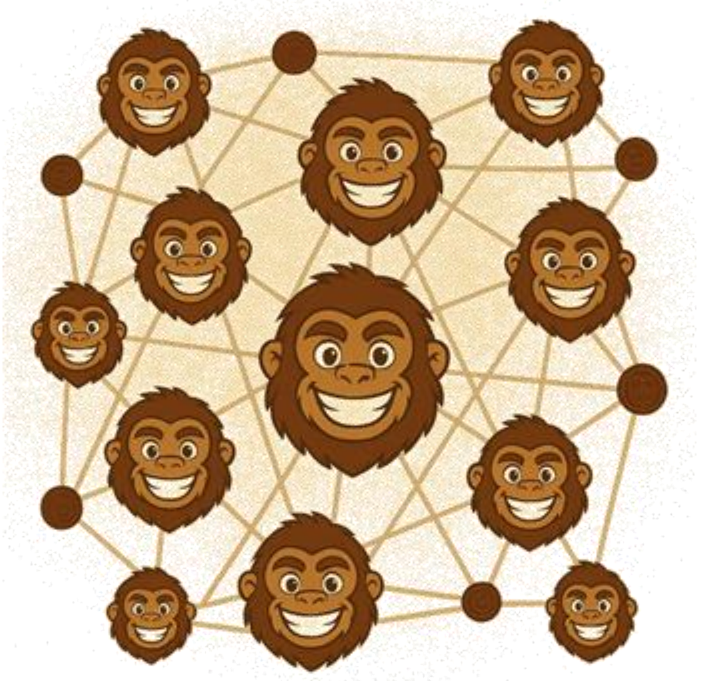
- *Impactful break-throughs will rarely be achieved by using default configurations*

Recommendations

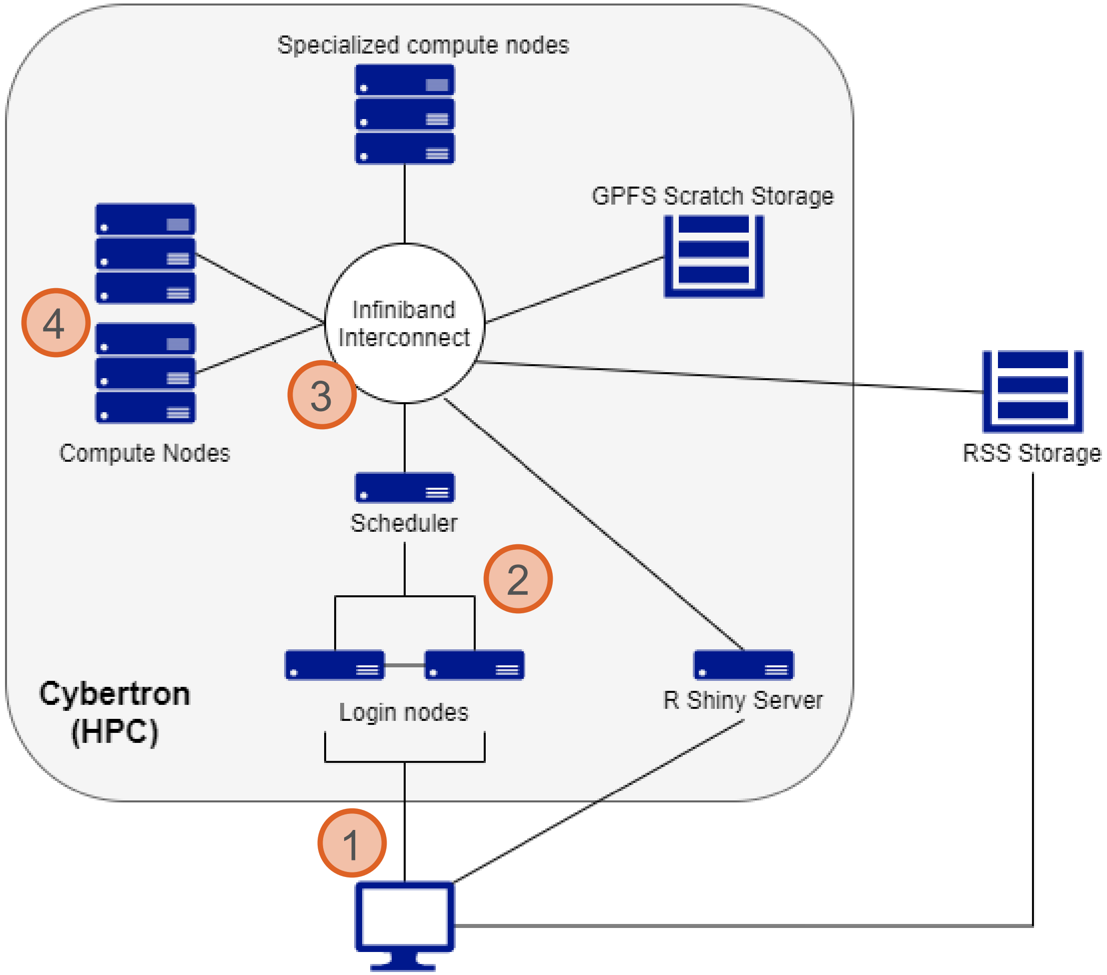
- Make the investments to upgrade your people, not just their technology
  - Move to open-source, programmatic workflows
  - Learn to batch your jobs
  - Take advantage of GPUs
  - Invest time to leverage orchestration tools such as Snakemake or Nextflow
- Partner with data and computer scientists who know how to code and/or achieve maximum parallelism
- AI will enhance your work, but it can’t replace you (yet)
  - Leverage AI as a workflow enhancer and automater

It’s a graph not a tree

Understanding the true shape of research



How an HPC works



1. User logs in to the system

2. User submits a job request
   1. Specify resources required
   2. Specify commands to run

3. The schedular waits for available resources and dispatches the job

4. Compute nodes perform the job.

5. Repeat (potentially 100s – 1000s of jobs)

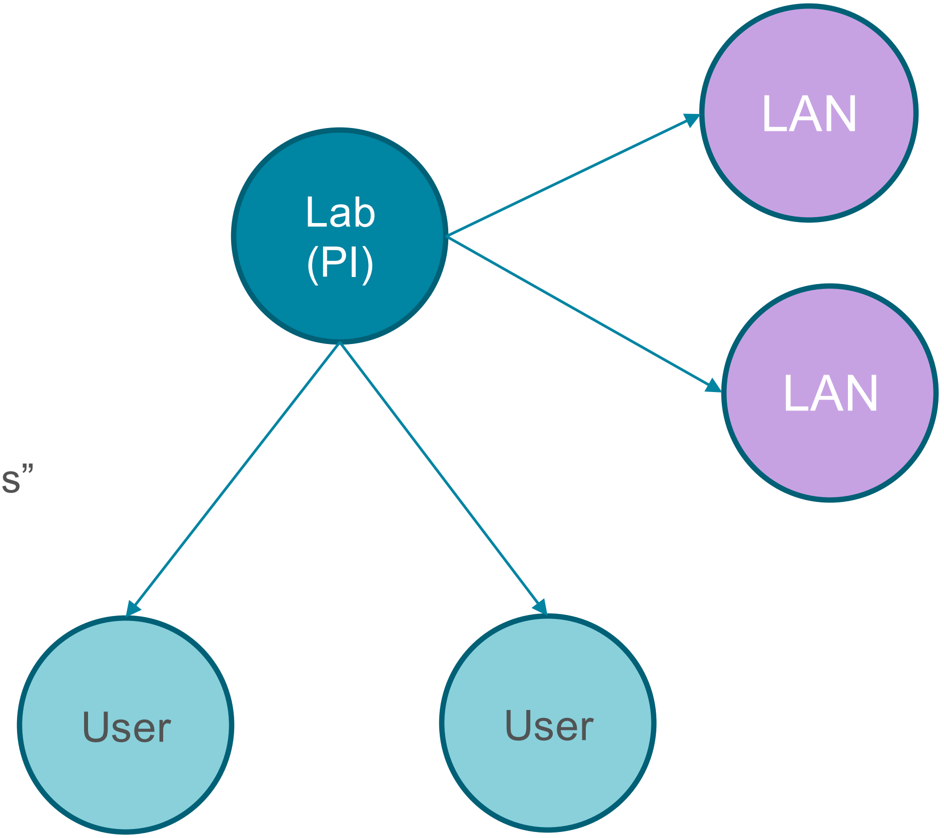
iLab

- iLab assumes a hierarchical model (tree)
  - Labs have a PI
  - Accounts assigned to a single lab
  - Users can (in theory) be part of multiple labs

- iLab is built on a self-service model
  - Users select Lab and account
  - Users request a service and quantity
  - Users consume the resource

- iLab interface ensures
  - Users do not need to know account numbers, just the “pretty names”
  - Accounts are assigned to the right lab
  - Users are aware of expiration dates

- iLab requires
  - All users are registered in iLab and assigned a lab
  - Accounts are associated with 1 and only 1 Lab
  - Labs have 1 and only 1 PI

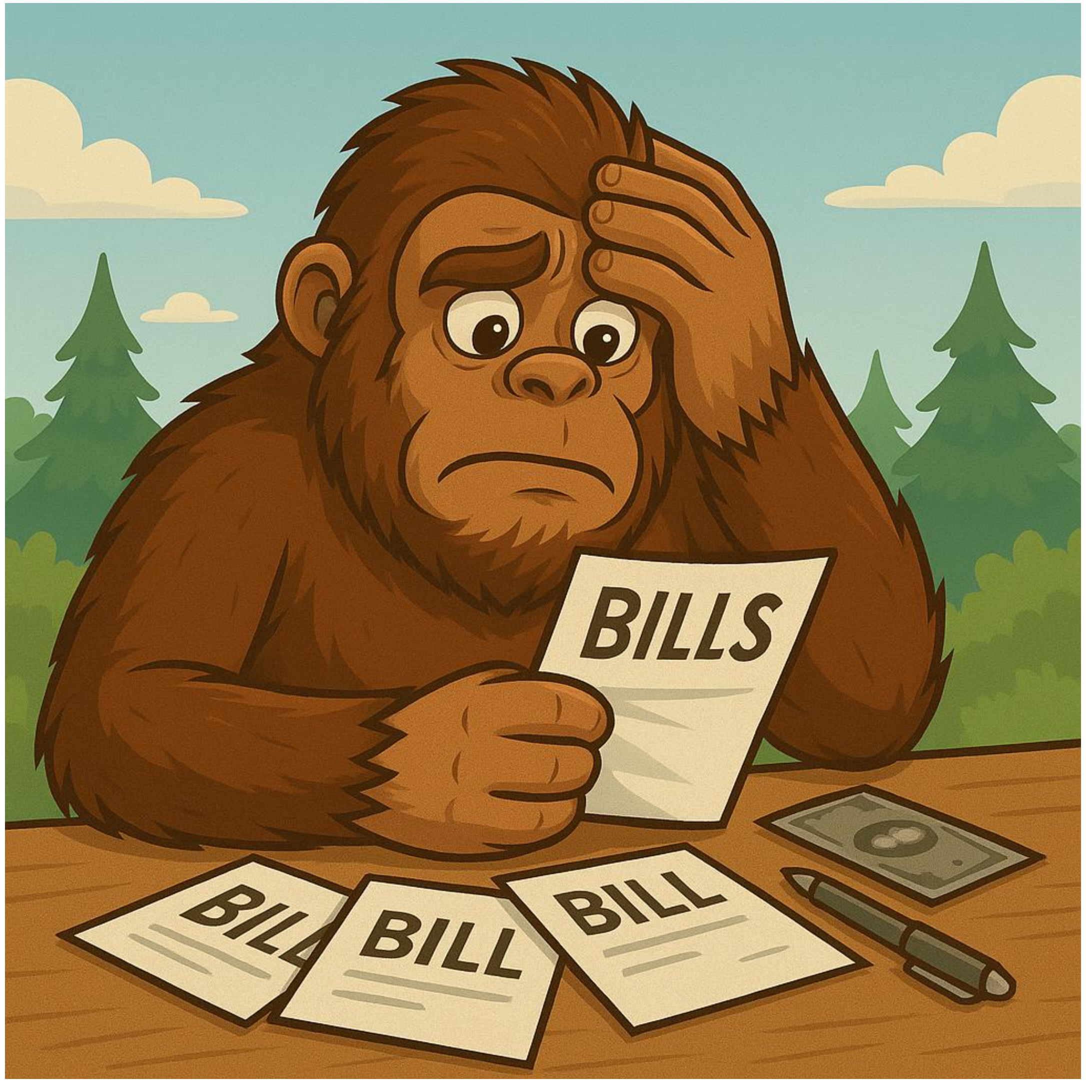


Billing challenges

- Self-service billing does not map well to the self-service consumption model for HPC
  - iLab is transactional like a grocery store
  - HPC is utility like your electricity
  - Reactive billing is better

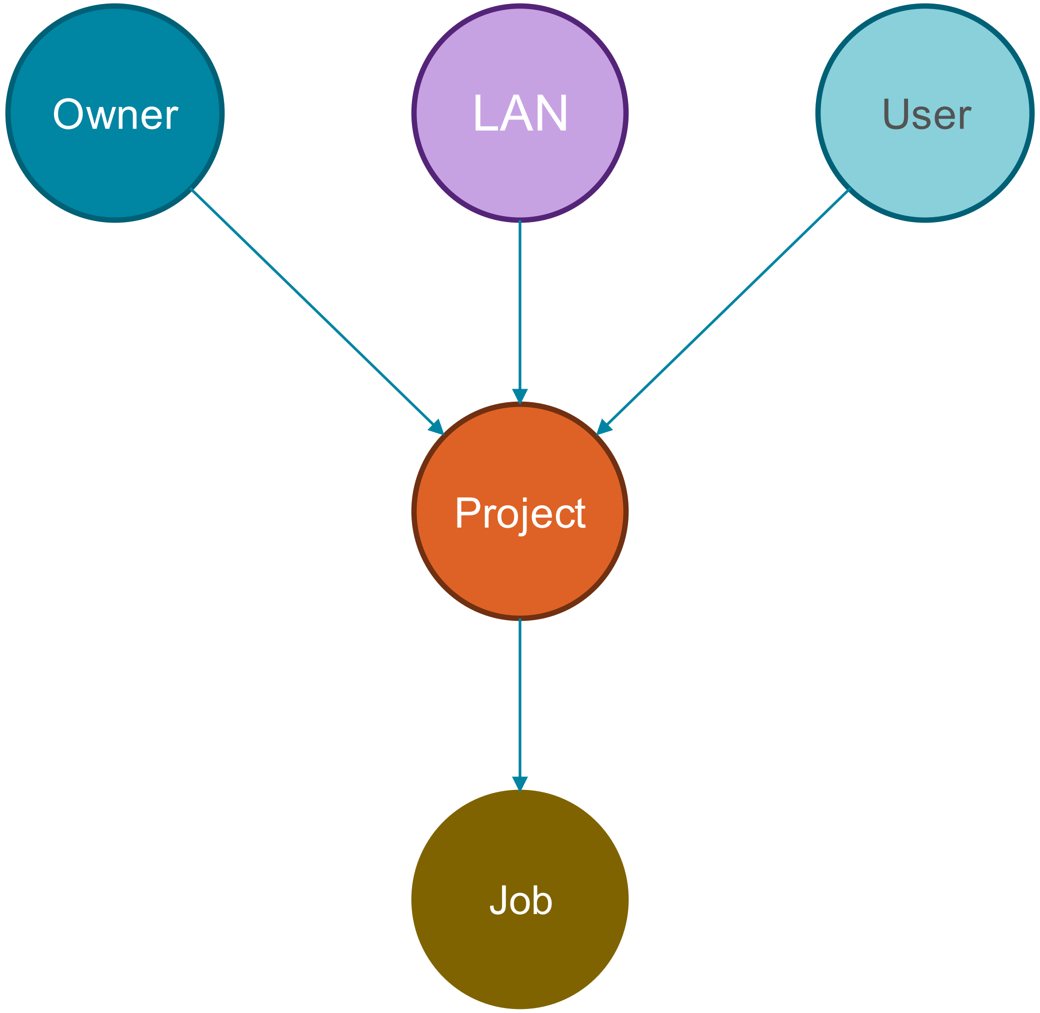
- Reactive billing requires we enter charges on behalf of users
  - Requires knowledge of constructs that exist only in iLab
  - “Lab” is not a defined entity
  - “PI” is not a defined entity
  - Not all users belong to labs, PIs, or even to the institution
  - Collaborations across labs and groups

- Account management is messy
  - PIs have many grant accounts
  - Labs have multiple projects and different people working on them
  - PIs charge to accounts that do not “belong” to them
  - Accounts expire and are replaced frequently



Project oriented work

- A “project” connects jobs to a grant or activity
- Minimally defined with
  - A project name
  - An owner
  - A list of users
  - Billing info
    - Account
    - AU or department code
- Project code
  - Each project assigned a code
  - Tags each job to link to project
  - Single code supplies all project metadata for billing, tracking and reporting



Advantages of Project oriented work

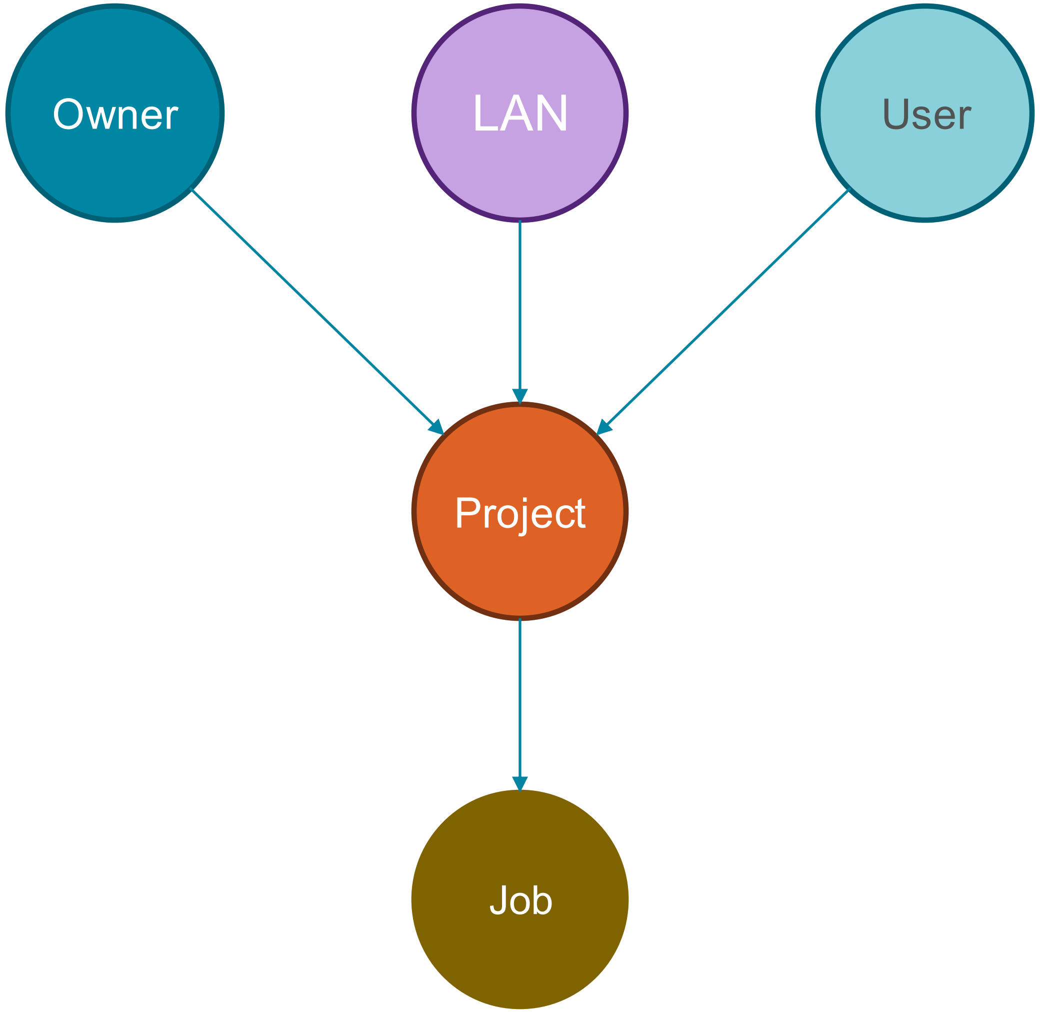
- Billing metadata is abstracted so users don't have to keep track of it

- Users have a stable code to tag jobs
  - LANs and Owners can be updated behind the scenes
  - Multiple LANs can be accommodated

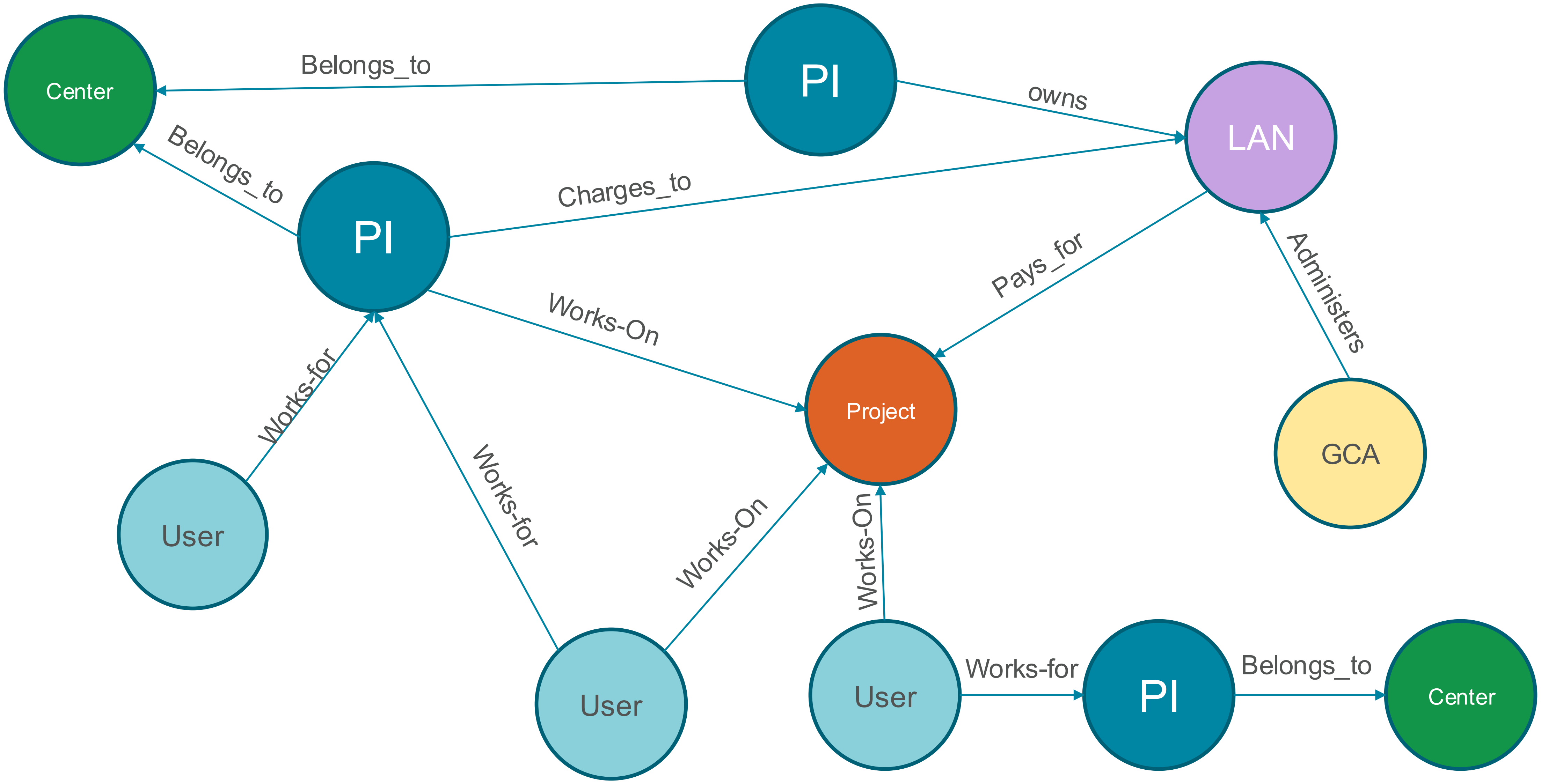
- Projects and collaborations are self-defined
  - Allows for easier collaboration across teams, departments, and divisions

- Project attributes can be used to manage service
  - Control access and permissions to charge to certain grants
  - Put guardrails on resources and costs consumed by specific projects
  - Can manage access to hardware and adjacent resources
  - Much of the collection and validation can be automated

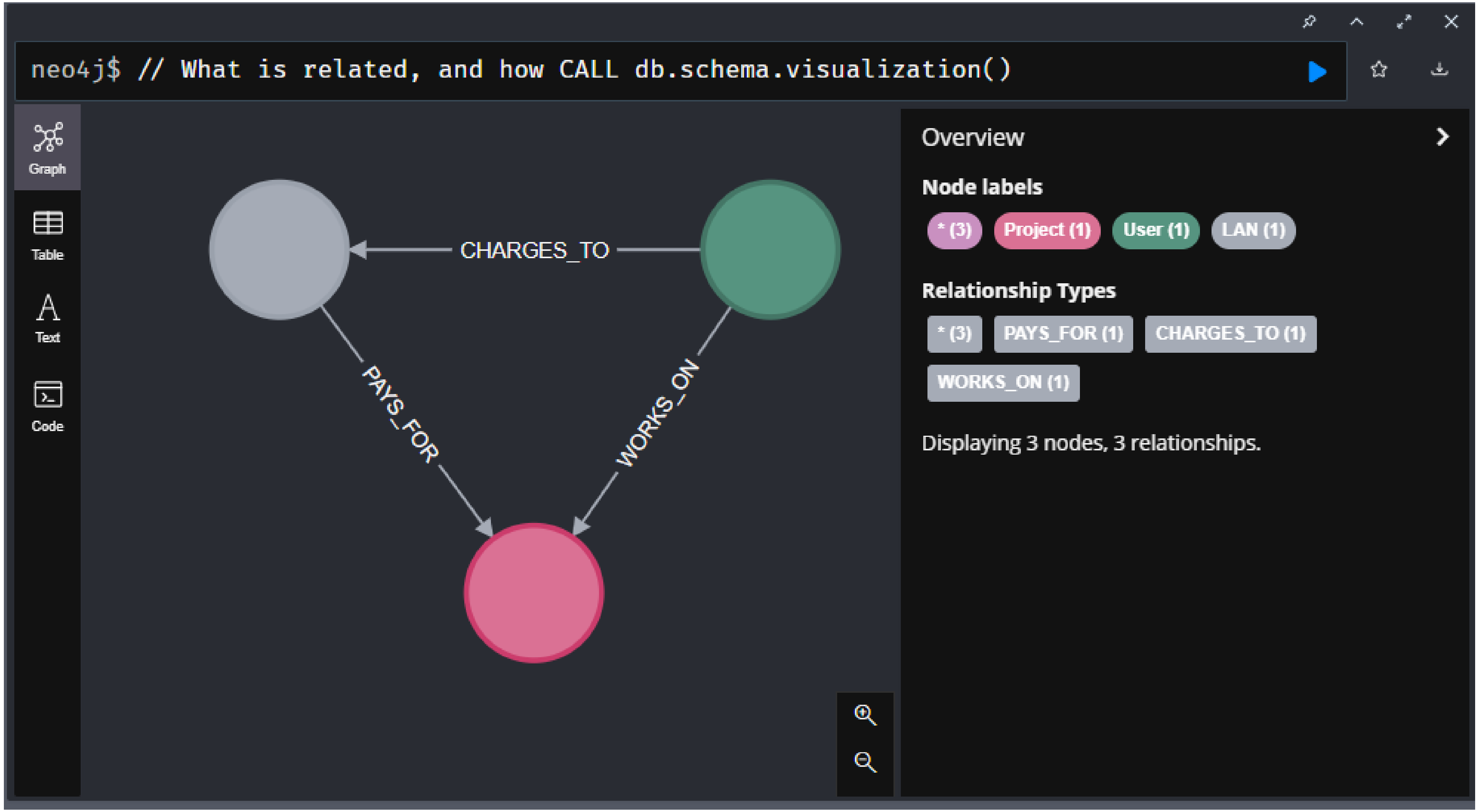
- Projects can contain virtually unlimited metadata
  - Useful for BI



Things we learned



Intelligence beyond billing



```
neo4j$ // What is related, and how CALL db.schema.visualization()
```

Overview

Node labels

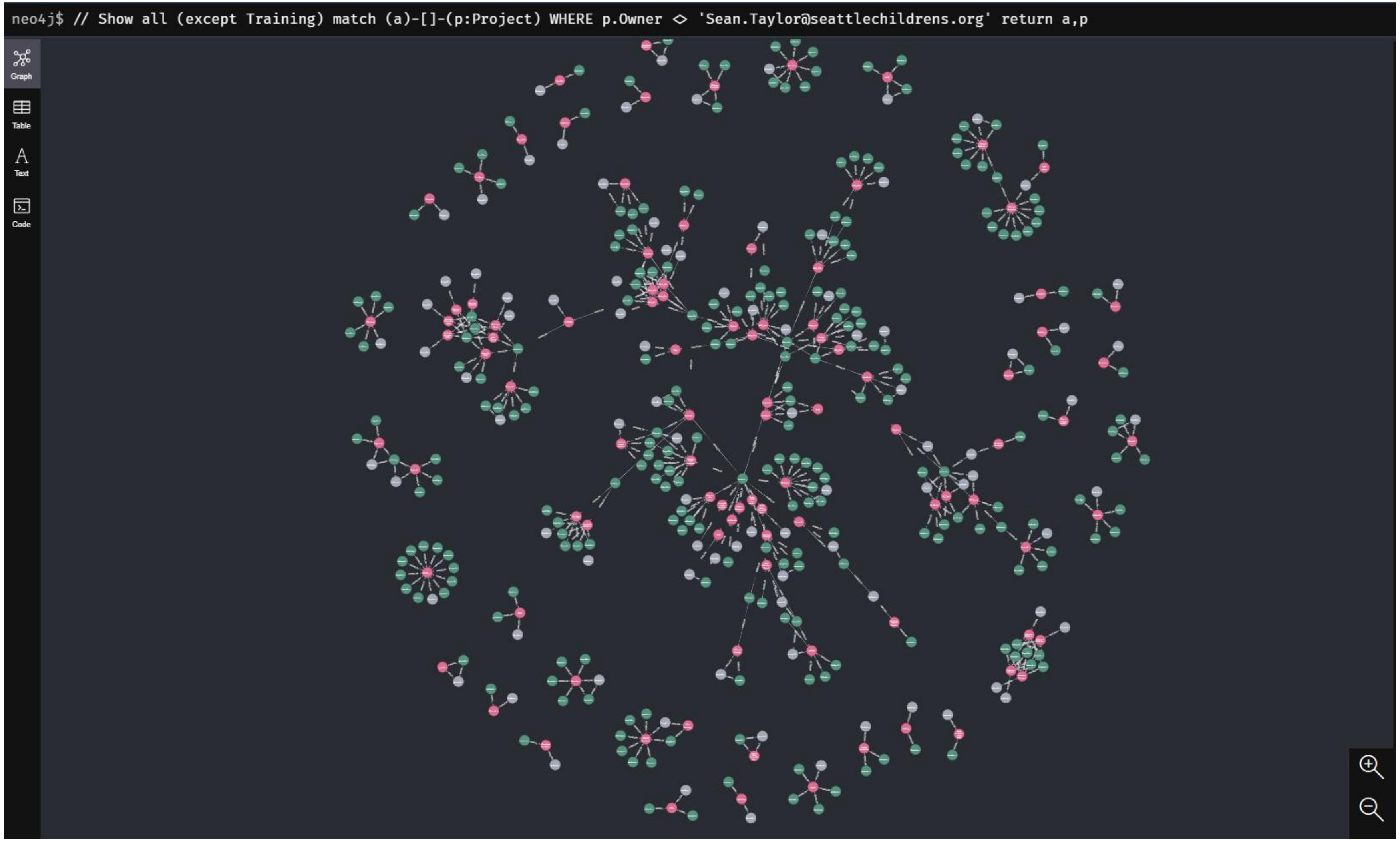
* (3) Project (1) User (1) LAN (1)

Relationship Types

* (3) PAYS_FOR (1) CHARGES_TO (1) WORKS_ON (1)

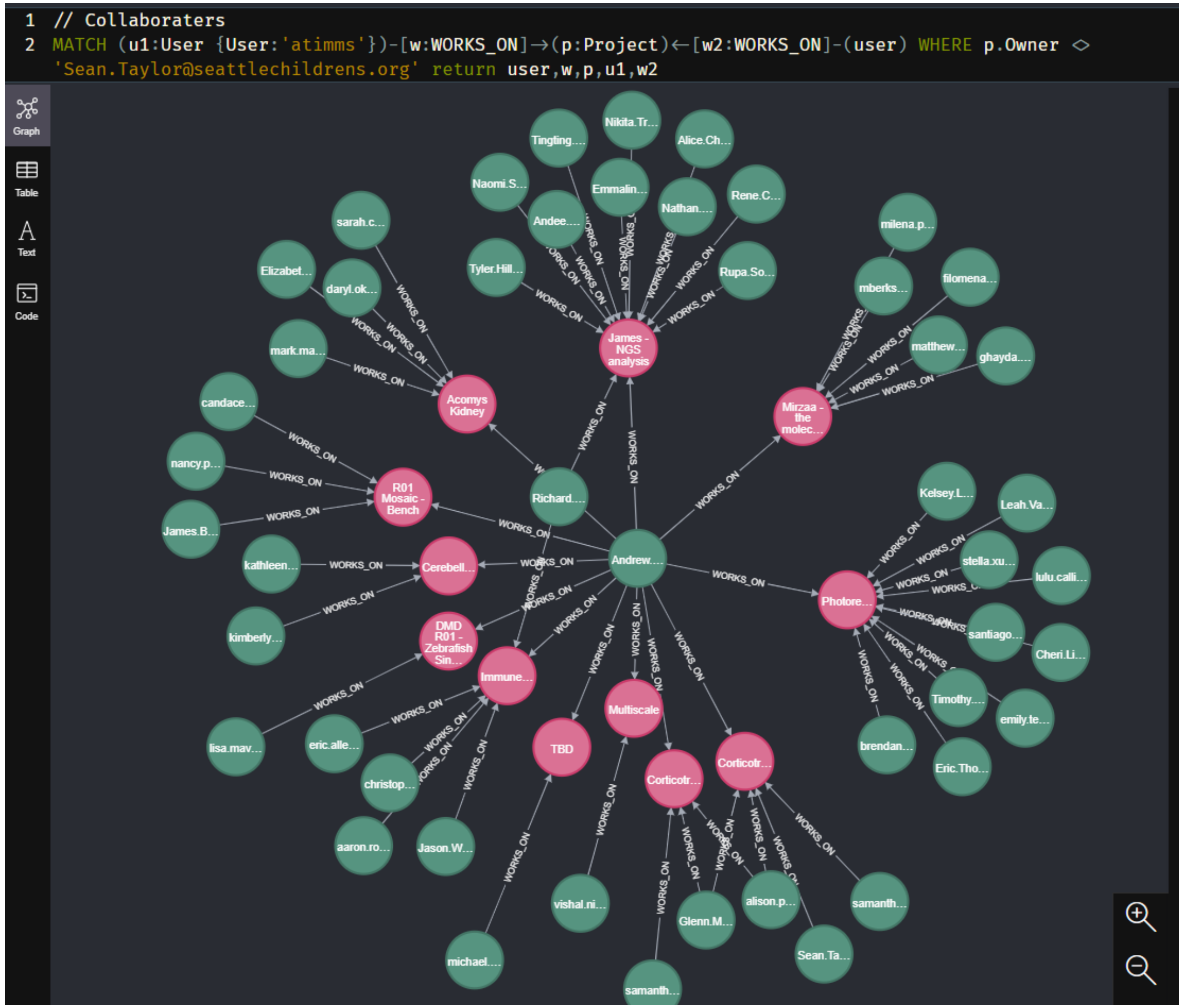
Displaying 3 nodes, 3 relationships.

Mapping collaborative relationships



```
neo4j$ // Show all (except Training) match (a)-[]-(p:Project) WHERE p.Owner <> 'Sean.Taylor@seattlechildrens.org' return a,p
```

Individual collaboration space



```
// Collaboraters
MATCH (u1:User {User:'atimms'})-[w:WORKS_ON]→(p:Project)←[w2:WORKS_ON]-(user) WHERE p.Owner <>
'Sean.Taylor@seattlechildrens.org' return user,w,p,u1,w2
```

Institutional business intelligence

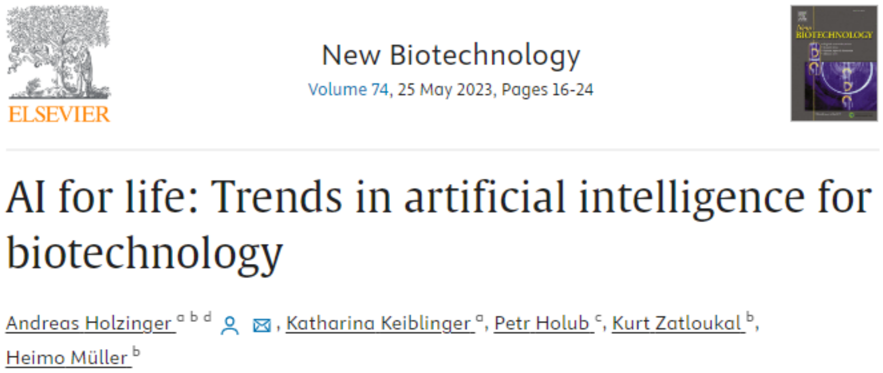
- A true map of relationships within Institute
- For cores and services
  - Identify areas of high connectivity
  - Quantify the business value of services
  - Identify key players in domain space
  - Identify areas of intersectionality for investment
- For institute
  - Identify areas of cross center collaboration
  - Identify grants/PIs that are getting high leverage from grants
  - Identify key resources and choke points

Data Digitalization

Digitalization definition: when data from throughout the organization and its assets is processed through advanced **digital technologies**, which leads to **fundamental changes** in business processes that can result in **new business models and social change.**

AI/ML in biotechnology

It is important to emphasize at the very beginning that all the successful AI we are describing today relies entirely on digital technology to function. Digitalization is therefore the very first step towards any AI application.



New Biotechnology

Volume 74, 25 May 2023, Pages 16-24

AI for life: Trends in artificial intelligence for biotechnology

Andreas Holzinger [a b d], Katharina Keiblinger [a], Petr Holub [c], Kurt Zatloukal [b], Heimo Müller [b]

https://doi.org/10.1016/j.nbt.2023.02.001

Acknowledgements

Glenn Morton

Sebastian Smith

Bronson Petterson

Amy Leonardson

Lindsay Clark

Marc Carlson

Yeji Bae

Neerja Katiyar

Appendix

IDEs with Posit Workbench

- Notebooks in R Studio and Jupyter facilitate literate programming for templates, documentation, workflows
- Build code-embedded reports, books, presentations
- Share and deploy reports, interactive plots, apps using Shiny and Posit Connect
- Multilanguage development environments for writing code, building pipelines, developing packages and apps

RStudio®

jupyter

VS Code

Positron

Shiny

Containerization

- Reproducibility and portability: Containers bundle all necessary software, libraries, and dependencies, ensuring consistent execution across different HPC environments
- Simplified software management: Users can easily deploy pre-configured environments without requiring administrator privileges for software installation. Users can also build and deploy custom containers
- Improved resource utilization: Containers alleviate the complexities of installing and managing diverse bioinformatics tools and their specific dependencies, which can often conflict.
- Collaboration and sharing: Container images can be easily shared among researchers, enabling seamless collaboration and dissemination of bioinformatics pipelines and analysis environments.

docker

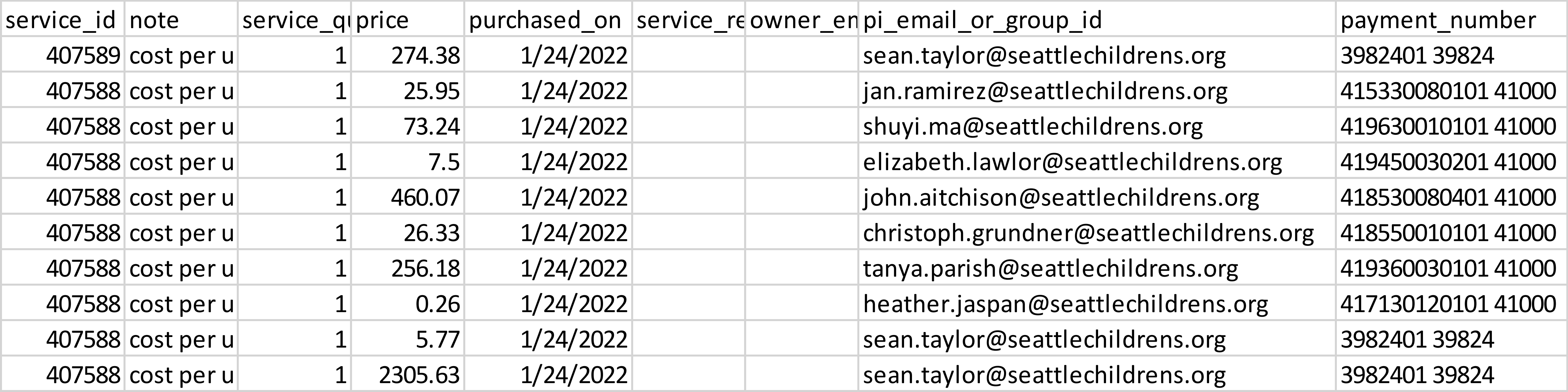
APPTAINER

The future of Scientific Computing Platforms

- Tighter integration with cloud
- High performance interactive computing platforms
- Adoption of GPU-accelerated workflows
- Moving from GPUs to TPUs
- Fully supported platforms with agentic AI tool suites

iLab solution

- Reactive billing based on actual usage
- Aggregate usage per LAN and bill monthly
- Submit monthly charges using iLab csv bulk upload
- Manually handle errors



service_id	note	service_qı	price	purchased_on	service_re	owner_en	pi_email_or_group_id	payment_number
407589	cost per u	1	274.38	1/24/2022			sean.taylor@seattlechildrens.org	3982401 39824
407588	cost per u	1	25.95	1/24/2022			jan.ramirez@seattlechildrens.org	415330080101 41000
407588	cost per u	1	73.24	1/24/2022			shuyi.ma@seattlechildrens.org	419630010101 41000
407588	cost per u	1	7.5	1/24/2022			elizabeth.lawlor@seattlechildrens.org	419450030201 41000
407588	cost per u	1	460.07	1/24/2022			john.aitchison@seattlechildrens.org	418530080401 41000
407588	cost per u	1	26.33	1/24/2022			christoph.grundner@seattlechildrens.org	418550010101 41000
407588	cost per u	1	256.18	1/24/2022			tanya.parish@seattlechildrens.org	419360030101 41000
407588	cost per u	1	0.26	1/24/2022			heather.jaspan@seattlechildrens.org	417130120101 41000
407588	cost per u	1	5.77	1/24/2022			sean.taylor@seattlechildrens.org	3982401 39824
407588	cost per u	1	2305.63	1/24/2022			sean.taylor@seattlechildrens.org	3982401 39824

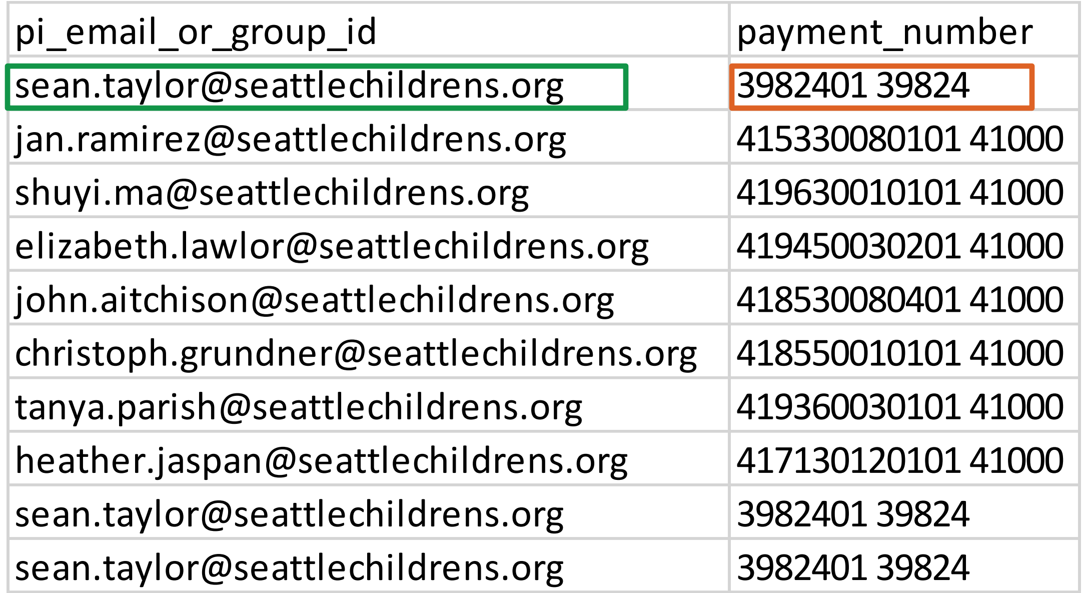
Billing requirements

- For billing in iLab, each job needs to be tied to
  - A user
  - A lab
  - Lawson Activity Number (LAN)

- Each line in the submission requires
  - A valid LAN+AU combination
    - Active LAN
    - Correct AU
    - Present in iLab system
  - Correct mapping between PI and payment_number

- There is no source of truth for this information!
  - At least not generally or easily or programmatically accessible
  - Rely on users to self-report the correct information

- There is too much metadata to tag to a HPC job



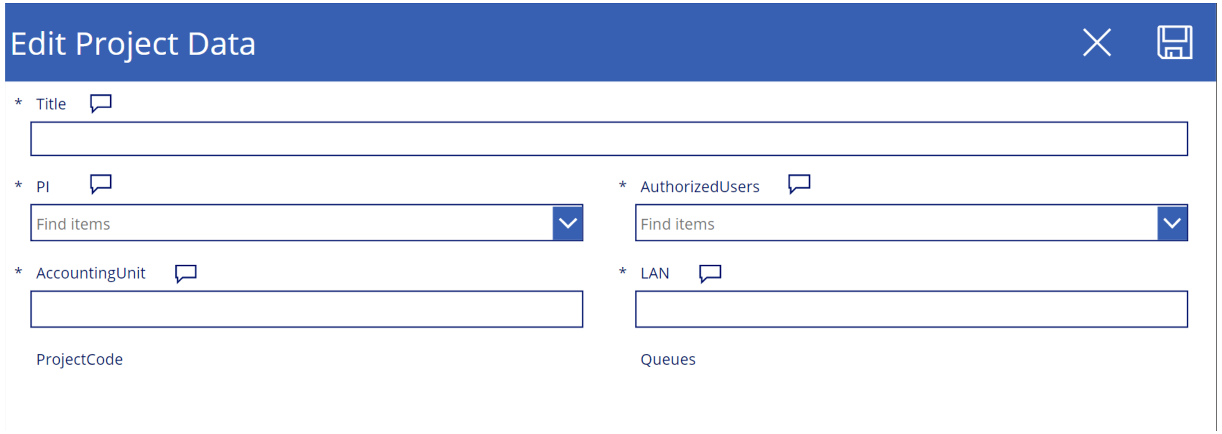
pi_email_or_group_id	payment_number
sean.taylor@seattlechildrens.org	3982401 39824
jan.ramirez@seattlechildrens.org	415330080101 41000
shuyi.ma@seattlechildrens.org	419630010101 41000
elizabeth.lawlor@seattlechildrens.org	419450030201 41000
john.aitchison@seattlechildrens.org	418530080401 41000
christoph.grundner@seattlechildrens.org	418550010101 41000
tanya.parish@seattlechildrens.org	419360030101 41000
heather.jaspan@seattlechildrens.org	417130120101 41000
sean.taylor@seattlechildrens.org	3982401 39824
sean.taylor@seattlechildrens.org	3982401 39824

Old registration process (Before 5/16/2022)

- Front end
  - SharePoint PowerApp
  - Maintained by RSC team

- Back end
  - Azure data tables
  - Managed by RSC team

- Workflow (~48 hours)
  - Manual checks for new or edited project data
  - Manual updates of Azure data tables
  - No approval process
  - No validation of data until monthly billing submission
  - Separate process for provisioning HPC access
- Generate Valid Bill (~24 hours)
  - Manual collection of logs
  - Manual log parsing and bill generation
  - Manual notification of expired billing info



Edit Project Data

* Title

* PI

Find items

* AuthorizedUsers

Find items

* AccountingUnit

* LAN

ProjectCode

Queues

Challenges (problems/annoyances)

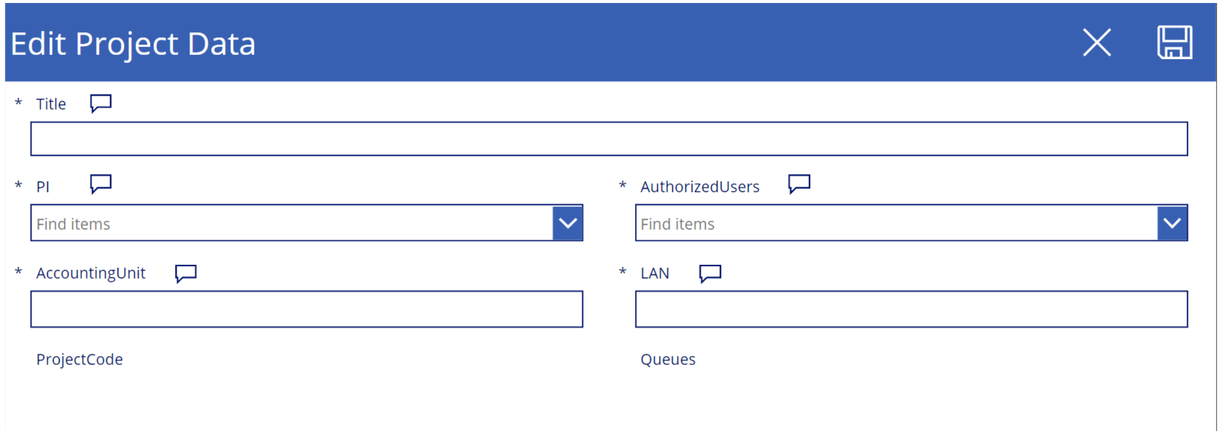
- Big problem with data quality
- Spent many hours fixing up names that were different between active directory and iLab
- LANs are expiring/expired
- LAN owner associations
- No established single source of truth
    - Sharepoint, several tables on Azure, iLab and Tableau

New registration process (5/16/2022 and later)

- Front end
  - SharePoint PowerApp
  - Maintained by RSC team

- Back end (THANK YOU WARREN!)
  - Postgres Database
  - Managed by Platform team

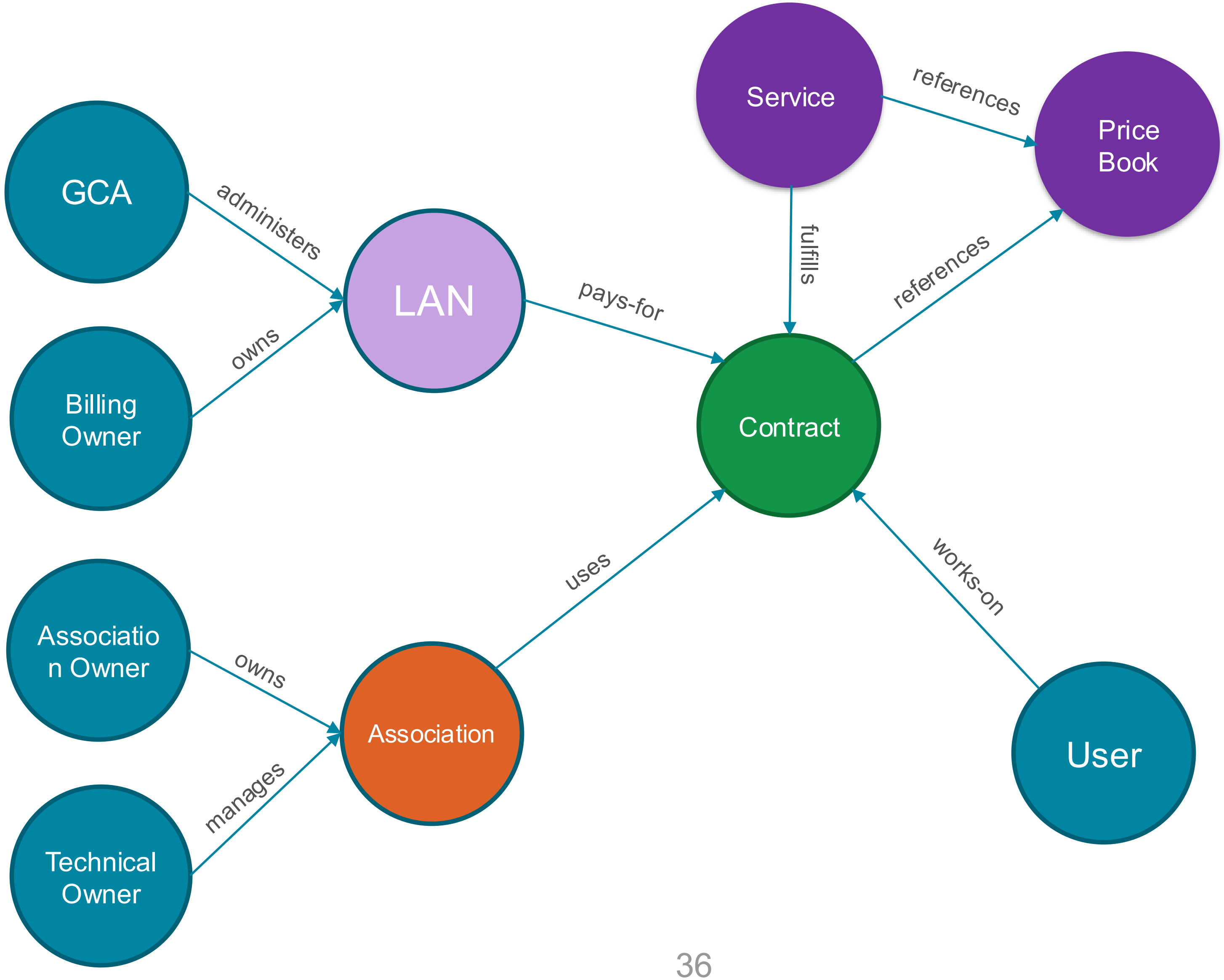
- Workflow (~5 minutes)
  - Automated checks for new or edited project data
  - Automated updates of PostgreSQL data tables
  - Automated approval process
  - Automated validation of data on submission
  - Automated prompts for expiring LANs
  - Integrated process for provisioning HPC access through Jira
- Generate Valid Bill (~5 seconds)
  - Automated collection of logs
  - Automated log parsing and bill generation
  - Automated notification of expiring billing info



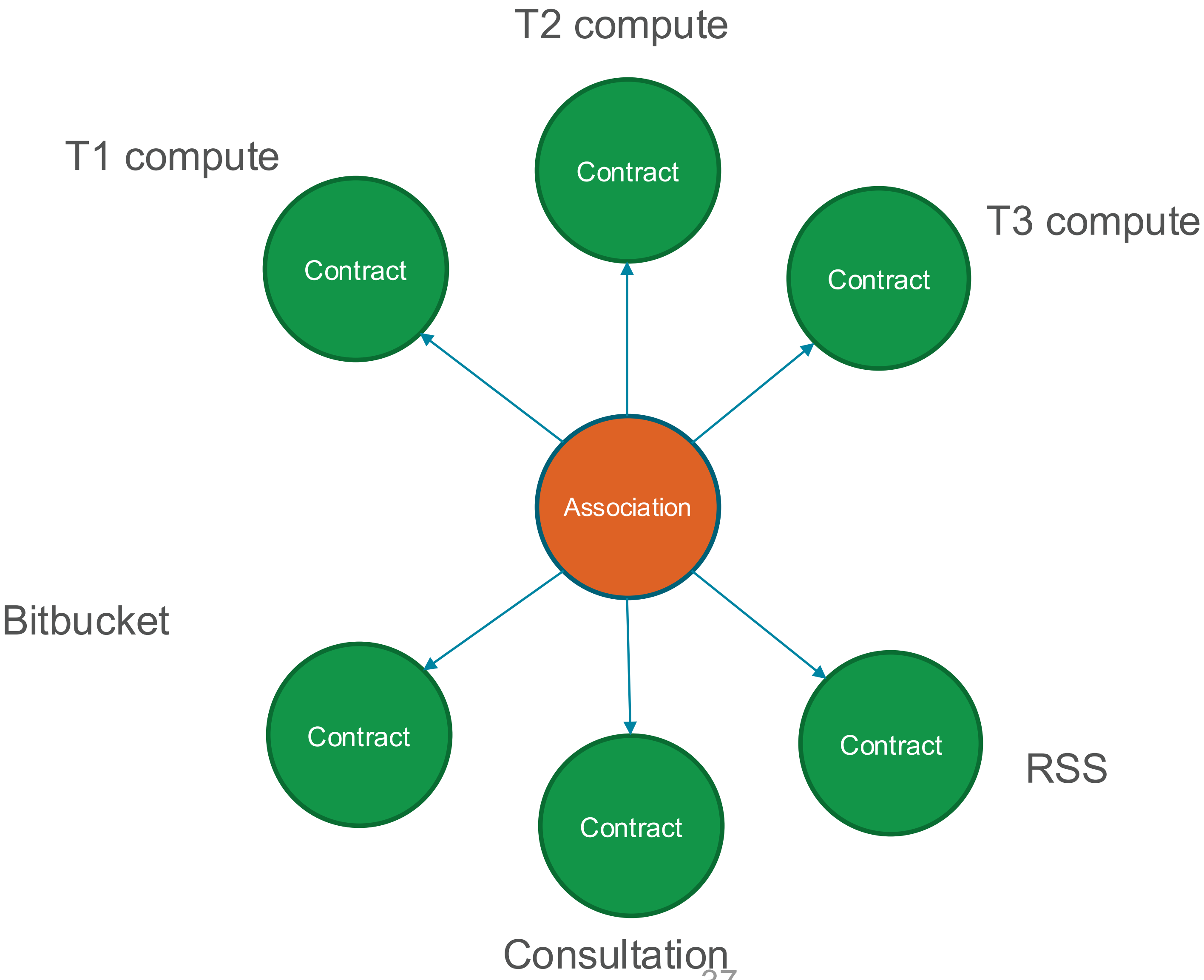
Edit Project Data

* Title	
* PI: Find items	* AuthorizedUsers: Find items
* AccountingUnit	* LAN
ProjectCode	Queues

Enhanced node topography



Association-contract model


T2 compute
T1 compute
T3 compute
Contract
Contract
Contract
Association
Bitbucket
Contract
Contract
Contract
RSS
Consultation