Real Databases in Astronomy

Ani Thakar
JHU Institute for Data Intensive Engineering and Science (IDIES)
LSSTC-DSFP Workshop, March 2019

- Case studies of SDSS and Pan-STARRS
  - Catalog archives for both designed at JHU/IDIES
  - Both used the same DBMS platform
- SDSS: a single monolithic database
  - Single instance deployed entirely on a single DB server
- Pan-STARRS: a distributed database
  - Single instance distributed across cluster
- Lessons from both
- Looking ahead to WFIRST and LSST

Outline

Database Wishlist

Price (academic pricing)

Query Optimizer

- Absolutely critical
- Must handle a range of query patterns
- Meaningful query cost estimation

Native floating point support

Complex science schema support

- Efficient analysis built into the db

Customized help

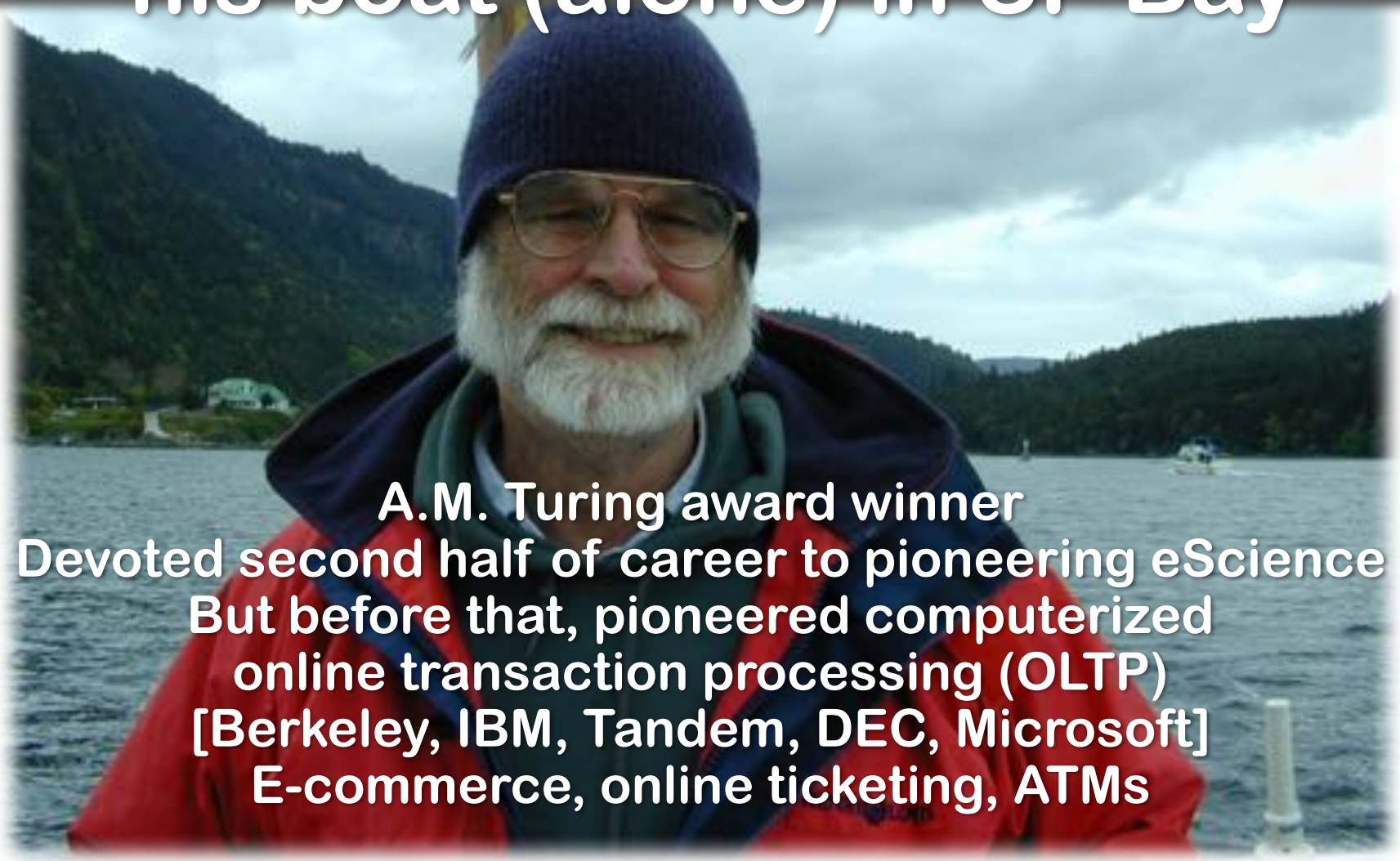
Microsoft SQL Server

- ✓ **Price (academic pricing)**
  - Much more affordable than Oracle, DB2
- ✓ **Query Optimizer**
  - One of the best in the business
- ✓ **Native floating point support**
- ✓ **Complex science schema support**
  - Compiled code in the db via SQLCLR
- ✓ **Customized help**
  - Certified genius to help us port our data

Jim Gray – Certified Genius

- Helped us port our EDR data to SQL Server (2001)
- Helped us code up first SkyServer & sqlLoader versions
- Advised us during our migration away from OODBMS
- Literally became an astronomer to code science schema with us (e.g. Neighbors and Match tables)
- Won over science community skeptical about DBs/MS
- Helped us devise a "poor man's HTM" – Zones
- Set up our SkyServer log database to track usage
- Used SDSS data to hone SQL Server products
- Predicted in early 2000s the sensor network revolution

Lost at sea in early 2007, while sailing his boat (alone) in SF Bay



A.M. Turing award winner
Devoted second half of career to pioneering eScience
But before that, pioneered computerized
online transaction processing (OLTP)
[Berkeley, IBM, Tandem, DEC, Microsoft]
E-commerce, online ticketing, ATMs

Additional DBMS Goodies in SQL Server

Good admin tools

- Suite of tools for DBAs
- Including query profiling, performance monitoring and tuning tools

Data compression

Schema evolution support

- For updates to schema in real time

Read-only mode (for OLAP)

- Minimal logging and locking

SQL Server Issues

Until recently, Windows only

- Linux version available now

No native distributed version

- Oracle has cluster version

CSV data ingest slow

- Native format ingest not well documented

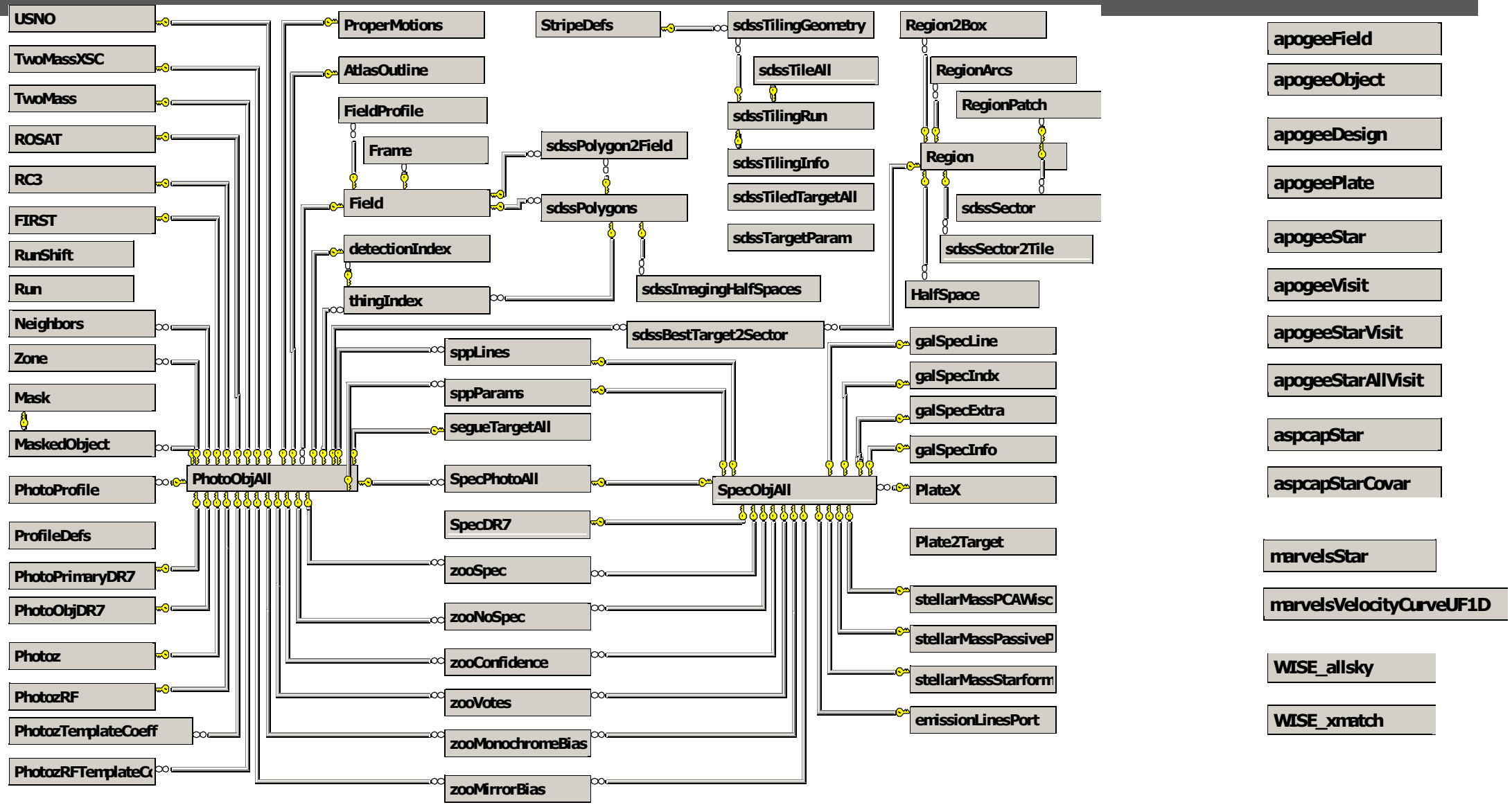
Had to do some things ourselves

- Had to develop data loading pipeline
  - sqlLoader

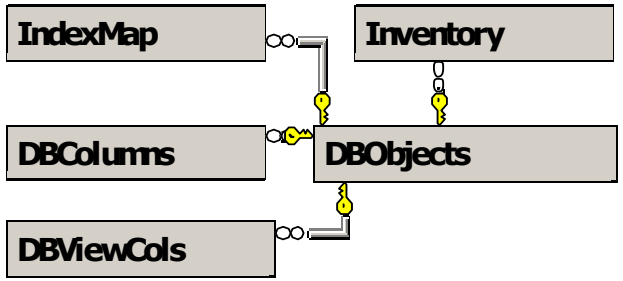
Discussion Items

- Do you think databases are the best option for large astronomical data sets going forward (LSST)?
- What are the one or two things that make them good?
- What new features would make them (even) better?
- What are the alternatives?
- What do you think the situation will be 10 years from now?

SDSS CAS Schema – Data Tables



SDSS CAS Schema – MetaData tables



PartitionMap

FileGroupMap

RecentQueries

QueryResults

Dependency

History

LoadHistory

PubHistory

DataConstants

SDSSConstants

Rmatrix

SiteConstants

SiteDBs

SiteDiagnostics

Diagnostics

Versions

Table Name	# Records	Data (KB)	Indexes (KB)
dbo.PhotoObjAll	1,231,051,050	3,210,876,312	1,585,756,872
dbo.AtlasOutline	1,222,390,340	1,125,957,960	1,683,672
dbo.Frame	3,752,184	1,037,872,848	267,984
dbo.PhotoProfile	44,595,857,733	780,798,208	2,101,392
dbo.Neighbors	25,610,066,940	572,702,576	1,990,048
dbo.WISE_allsky	563,921,584	460,866,240	124,548,584
dbo.SpecObjAll	4,851,200	218,133,024	1,602,704
dbo.PhotoObjDR7	364,857,538	104,935,528	4,208,016
dbo.PhotoPrimaryDR7	305,789,541	90,745,784	214,776
dbo.wiseForcedTarget	469,039,044	46,481,680	13,926,656
dbo.Zone	794,350,619	43,070,904	161,256
dbo.segueTargetAll	453,975,934	38,193,288	4,386,032
dbo.Photoz	208,474,076	34,579,536	82,168
dbo.USNO	253,732,084	20,661,224	49,432
dbo.ProperMotions	336,954,036	18,440,032	44,216
dbo.thingIndex	563,392,060	17,548,672	6,148,424
dbo.detectionIndex	932,886,174	12,487,448	11,388,920
dbo.apogeeObject	87,786,145	11,358,192	5,657,312

SDSS Data Table Sizes

- Tables whose size > 10 GB
- In descending order of size
- PhotoObjall largest table by far, most heavily indexed
- PhotoProfile has max rows but is thin and much less used
- Indices ~ 20% space

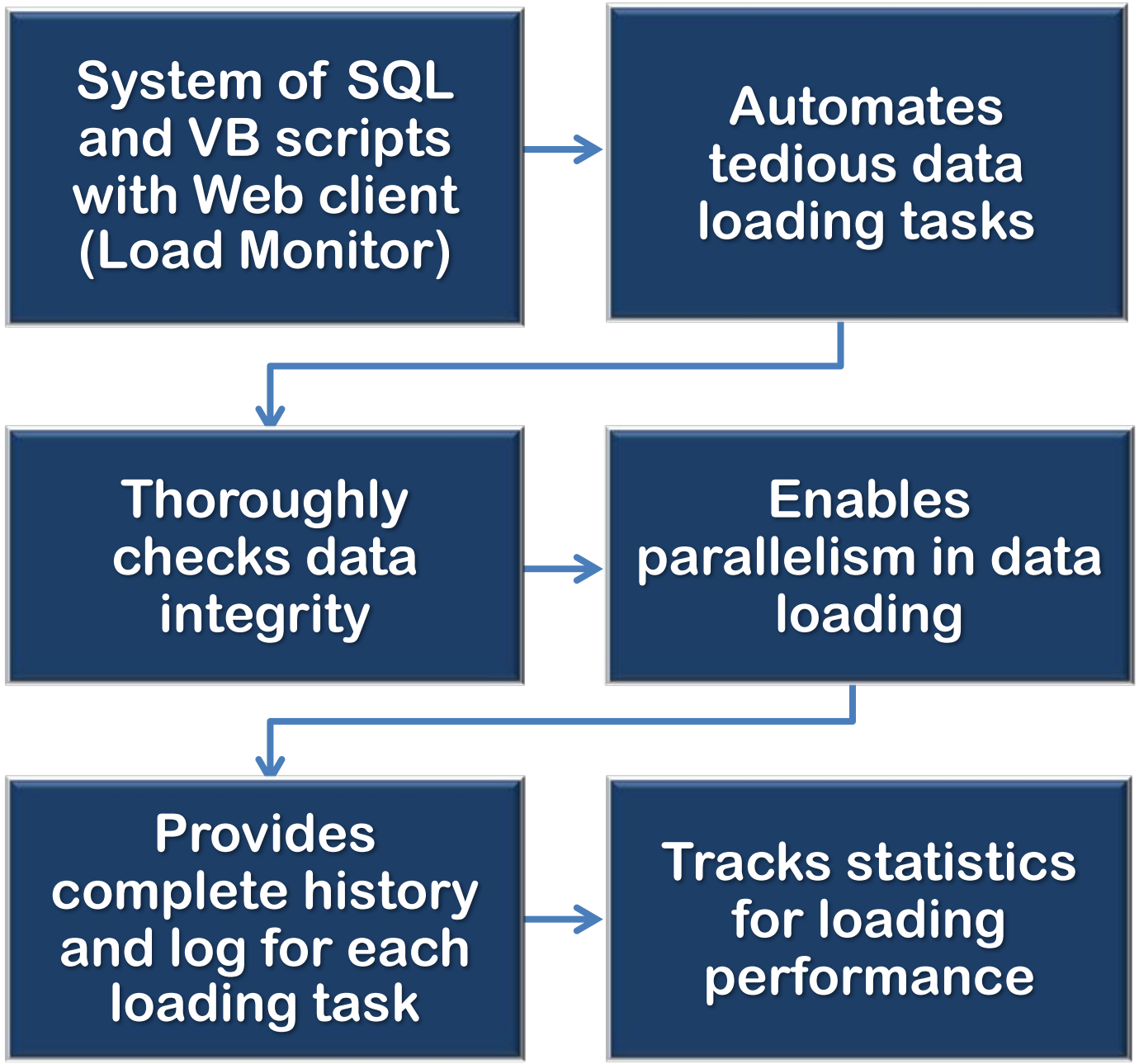
Data products that go into CAS

- Photo: SDSS, BOSS
- Spectro: SDSS, BOSS, SEGUE, APOGEE, MARVELS, MaNGA, eBOSS
- Survey Geometry: Window, Resolve, Region, Tiling etc.
- Xmatches: USNOB, 2MASS, 2DF, ROSAT, FIRST, WISE All Sky
- Galaxy Zoo 2 classifications

Ingested in CSV format by sqlLoader

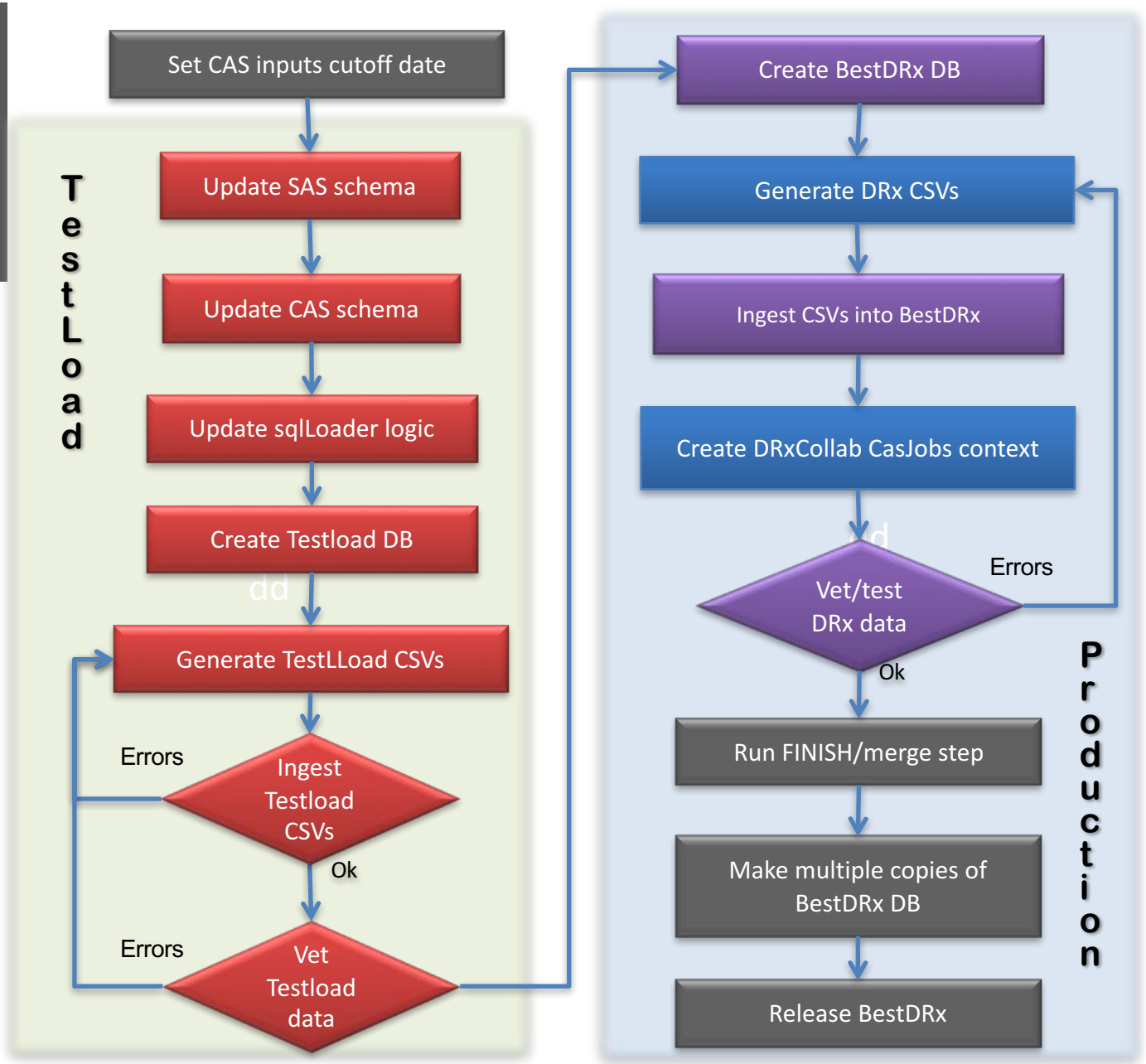
Data Loading

sqlLoader Data Ingest Pipeline

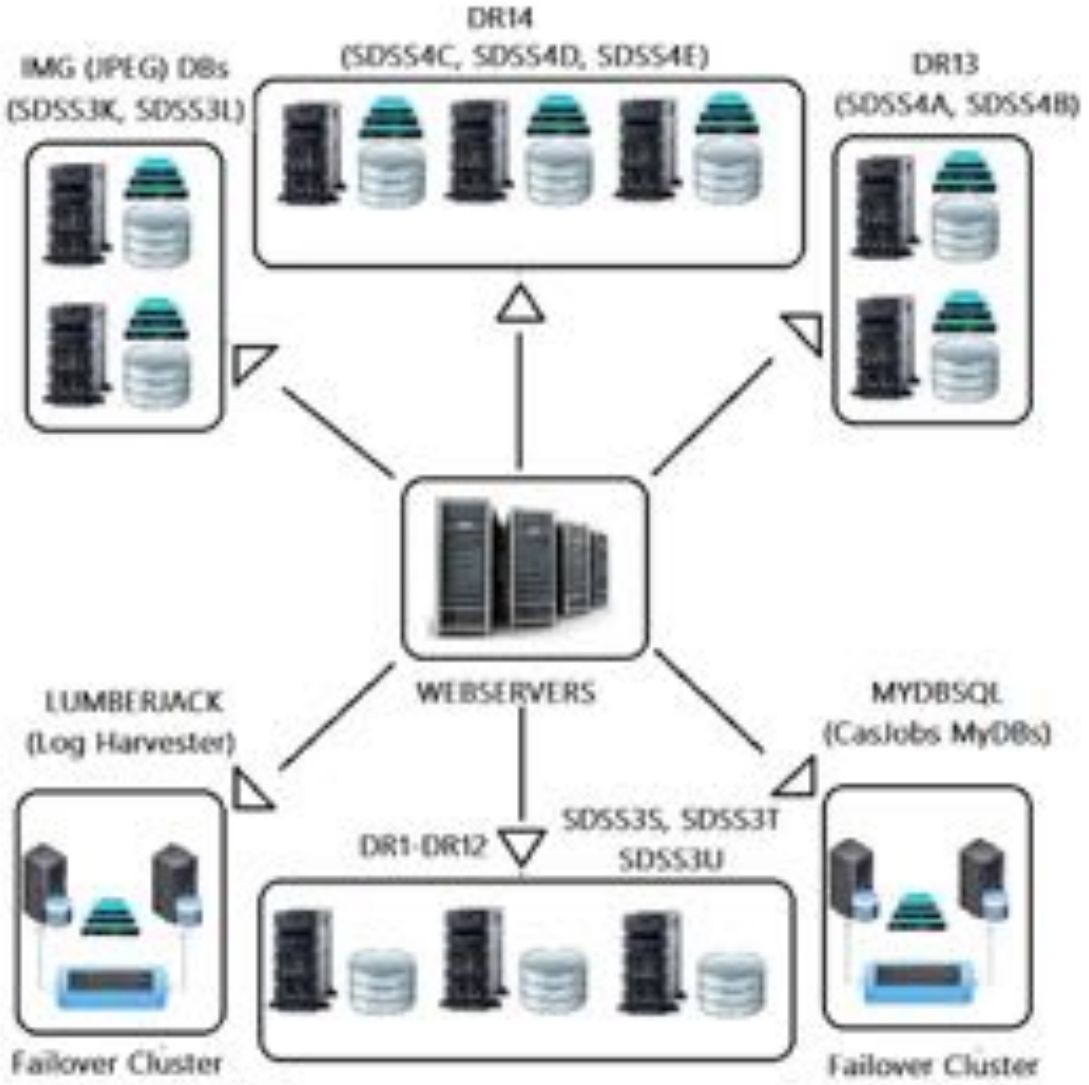


CAS Data Release Cycle

- Cycle must be repeated for each data product (photo, spectro, APOGEE etc.)
- Most steps involve multiple actions
- Testloads are usually a small subset of DR data
- Entire cycle can take ~ 2 months
- Objective is to give collab a few months preview of data
- Steps in gray are only done once when all data products are in



SDSS Cluster



IMG DBS

CPU: 2.46 GHZ DUAL CORE
MEMORY: 64GB
STORAGE: 6TB TOTAL (SSD)

DR13

CPU: 2.6 DUAL CORE 24 VIRTUAL CORES
MEMORY: 64GB
STORAGE: 18.6 TB TOTAL STORAGE (SSD)

DR14

CPU: 2.6 DUAL CORE 24 VIRTUAL CORES
MEMORY: 64GB
STORAGE: 1TB SSD STORAGE
AND 16TB SPINNING DISK

DR1 - DR12

CPU: 2.6 GHZ DUAL CORE 24 VIRTUAL CORES
MEMORY: 48GB
STORAGE: 100TB SPINNING DISK

LUMBERJACK

CPU: 2.46 GHZ DUAL CORE 24 VIRTUAL CORES
MEMORY: 256GB
STORAGE: 65TB SPINNING DISK AND 4TB SSDs

MYDBSQL

CPU: 2.46 GHZ DUAL CORE 24 VIRTUAL CORES
MEMORY: 256GB
STORAGE: 65TB SPINNING DISK AND 4TB SSDs

Lessons Learned from SDSS CAS

- **DBMS vendor choice is most important decision**
  - Look for one that will be around for a long time
  - Will keep up with industry standards and support you
  - Changing DB platform very hard later on
- **Minimize custom software development**
  - Use off-the-shelf wherever possible, follow standards
- **Log everything!**
  - Invaluable resource for your quarterly and annual reports
  - Assess how your tools and services are being used
  - Track errors and performance issues
  - Track responses to events like press releases, data releases
  - Analyze how scientists are adapting to new paradigm

Lessons Learned – DB Ops

- Size matters – data volume affects <u>everything</u>
  - Making a copy used to take a whole day! (now few hours)
  - Schema changes to largest tables impracticable
  - Rebuilding indices takes days!
- Need a plan to distribute copies to mirror sites
  - Dedicated site connected to fast pipes
- Keep enough copies online for:
  - Performance, high availability and load balancing
  - 6 copies needed for production data sets:
  - Smooth ops and development
  - Insure against disk failures!

Load-balancing	Development & Testing	Replication/ Distribution
3	2	1

Failed Disks Over 18 Years

- 1885 drives
- ~2.5%/year over 18 yrs
- 1.1 tons of HDD
- With SSD, hopefully much better going forward

Pan-STARRS

PSPS data (now hosted at STScI)

- Pan-STARRS Published Science Products
- ODM: Object Data Manager (JHU)
- SSDM: Solar System Data Manager
- PSI: PS Science Interface (front end)

ODM

- ~ 100 TB of data from multiple surveys
- 3π survey data set largest of them
- Medium-Deep Fields (MDFs), SS etc.

Pushing the Envelope

- The PS project came to us in an emergency
  - They had spent years and lots of money on their science database (ODM) design, with nothing to show for it
  - They needed to deliver something to consortium ASAP
- We spent 2 years (roughly 10 person-years) to extend and adapt our SDSS CAS design to Pan-STARRS
  - Really pushing the technology envelope
  - Monolithic single-server database no longer enough
  - Had to scale out for potential 100+ TB data set (25x)
  - Microsoft helped (provided an engineer and project mgr)

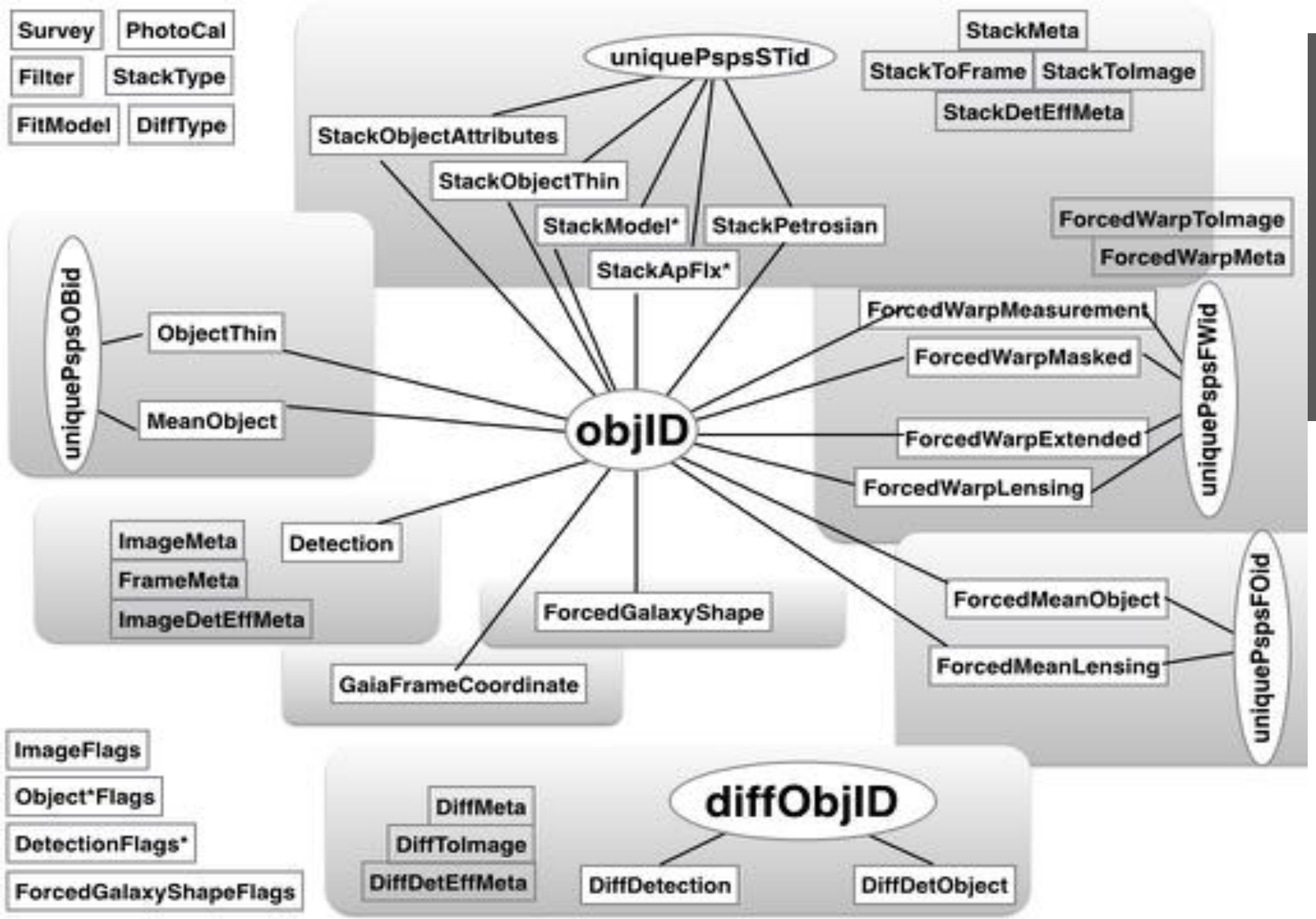
Distributed Design

- The query engine would still be CasJobs
  - It would connect to the head node that had Object table
  - Detection and StackDetection tables on cluster
- Detections would be distributed across 8 slice nodes
  - There would be 2 data slices on each node
  - There would be 2 versions of the 2 slices: Hot and Warm
- We would load data received in FITS format from IPP
  - Image Processing Pipeline
  - Convert data to CSV and load into the slices

Design Challenges

- Head node needed to access data on slices
  - Using Distributed Partition Views (DPVs)
  - SQL Server allows remote views across linked servers
  - This was largely untested technology in this regime!
- Weekly load/merge of data to release to collab.
  - Data released to consortium as soon as it was processed
  - This was a huge step up from our annual SDSS DRs
  - Required us to design complex distributed workflow
    - Dance of hot and warm slices being interchanged
    - Maintain performance, high availability and redundancy
  - Also required use of innovative SQL Server features to insert new data into existing very large tables
    - Use of the UPSERT feature

PS1 Schema


Survey
PhotoCal
Filter
StackType
FitModel
DiffType
uniquePspsSTid
StackMeta
StackToFrame
StackToImage
StackDetEffMeta
StackObjectAttributes
StackObjectThin
StackModel*
StackPetrosian
StackApFlx*
ForcedWarpToImage
ForcedWarpMeta
uniquePspsOBid
ObjectThin
MeanObject
ForcedWarpMeasurement
ForcedWarpMasked
ForcedWarpExtended
ForcedWarpLensing
uniquePspsFWid
objID
ImageMeta
FrameMeta
ImageDetEffMeta
Detection
ForcedGalaxyShape
GaiaFrameCoordinate
ForcedMeanObject
ForcedMeanLensing
uniquePspsFOid
ImageFlags
Object*Flags
DetectionFlags*
ForcedGalaxyShapeFlags
diffObjID
DiffMeta
DiffToImage
DiffDetEffMeta
DiffDetection
DiffDetObject

ODM System View

System Operation UI

System Health Monitor UI

Query Performance UI

System & Administration Workflows
Orchestrates all cluster changes, such as, data loading, or fault tolerance

Configuration, Health & Performance Monitoring
Cluster deployment and operations

Pan-STARRS Cloud Services for Astronomers

MS Windows Workflow Foundation
Tools for supporting workflow authoring and execution

Loaded Astronomy Databases
~70TB Transfer/Week

Deployed Astronomy Databases
~70TB Storage/Year

Query Manager
Science queries and MyDB for results

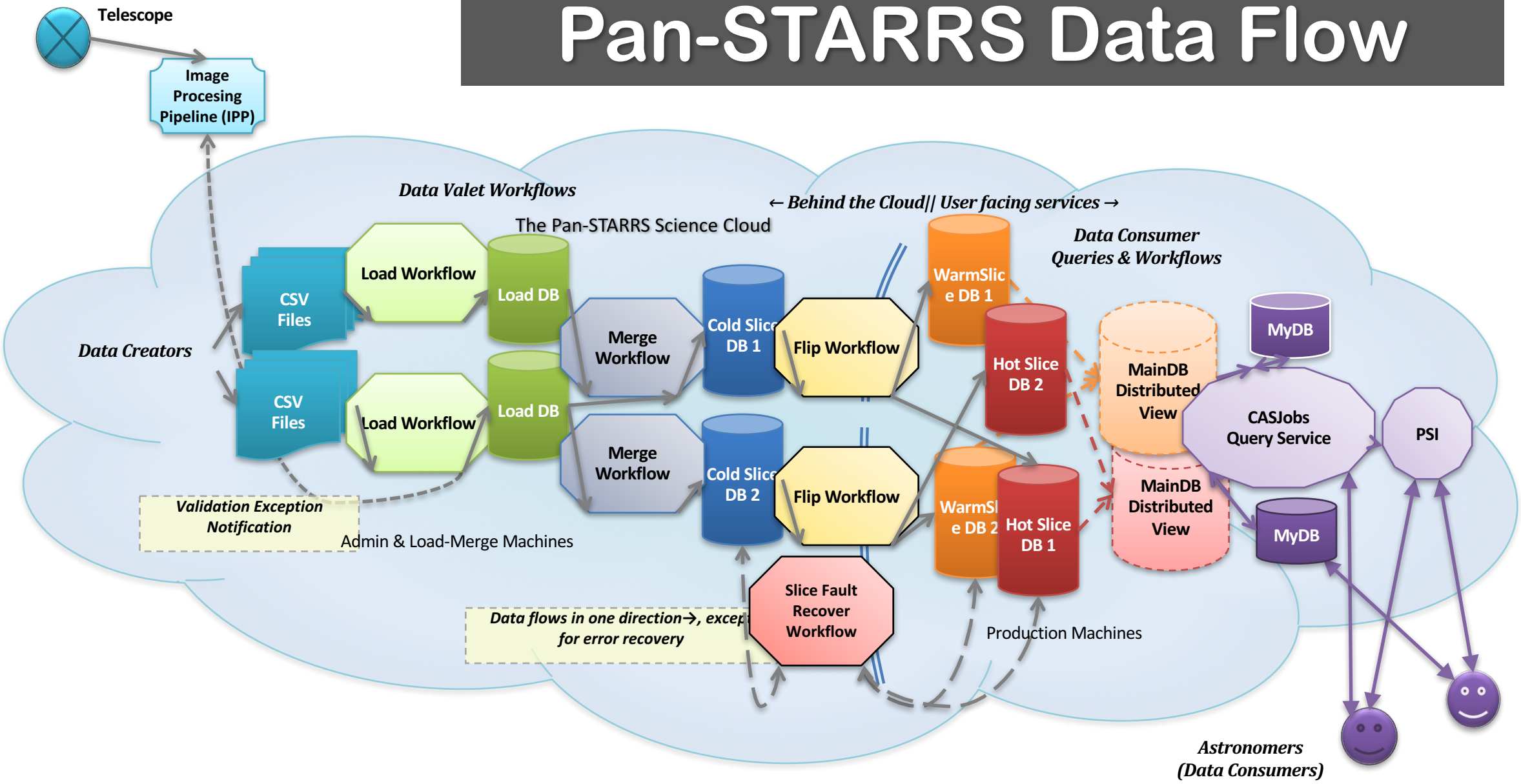
Pan-STARRS Telescope

Image Processing Pipeline
Extracts objects like stars and galaxies from telescope images
~1TB Input/Week

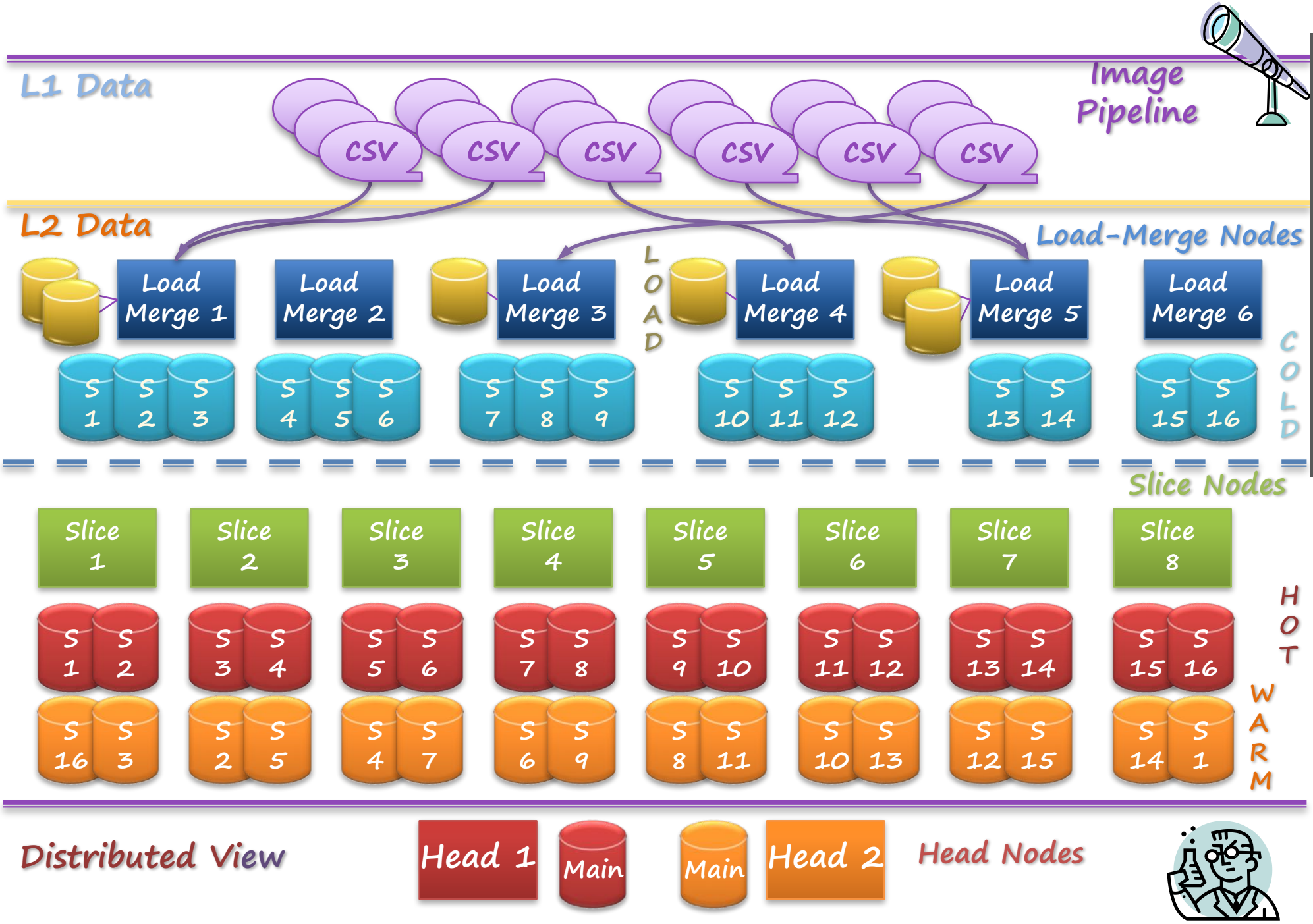
Data Flow

Control Flow

Pan-STARRS Data Flow

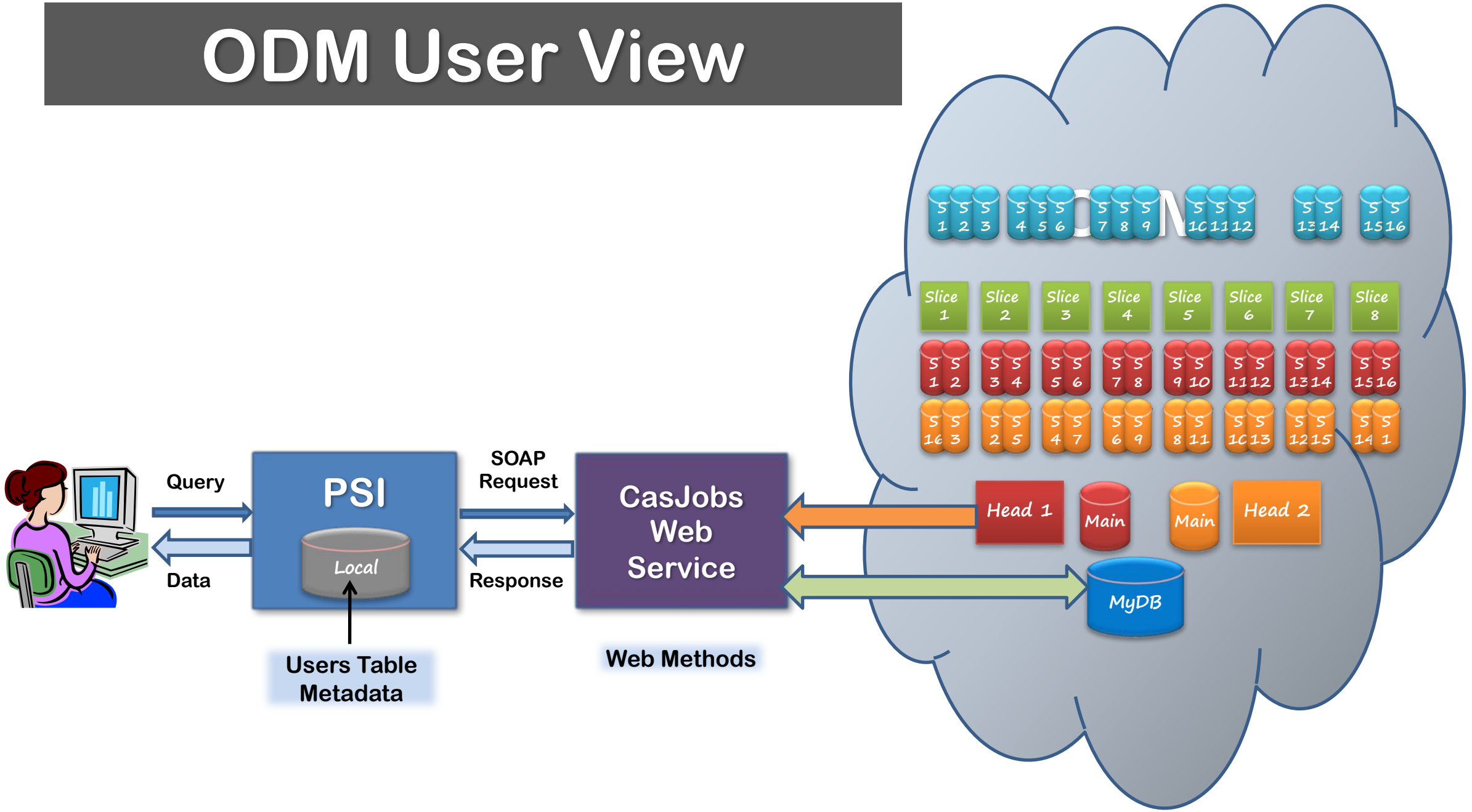


Distri-buted Data Layout



LSSTC-DSFP Workshop, Rutgers, March 2019

ODM User View

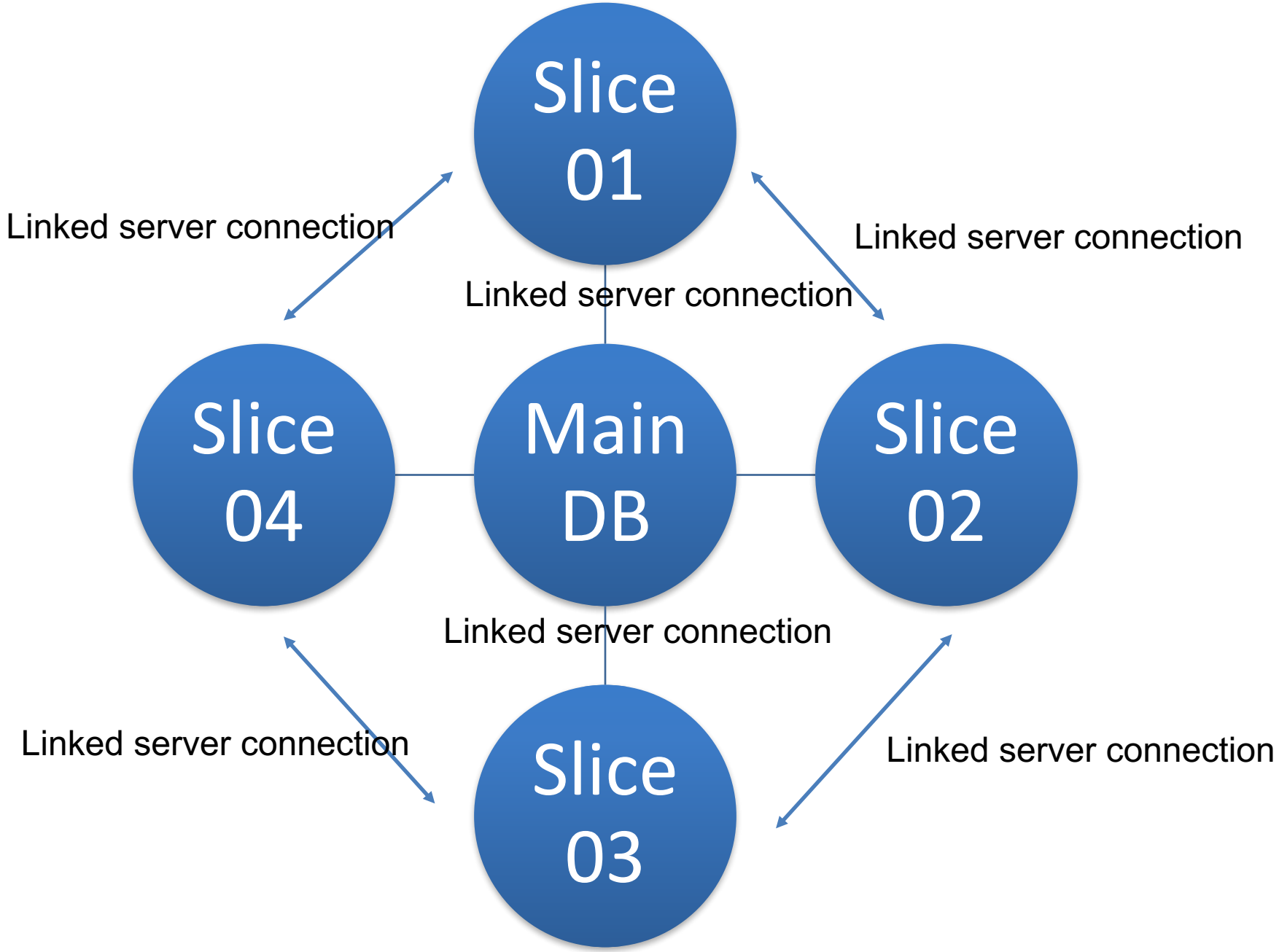


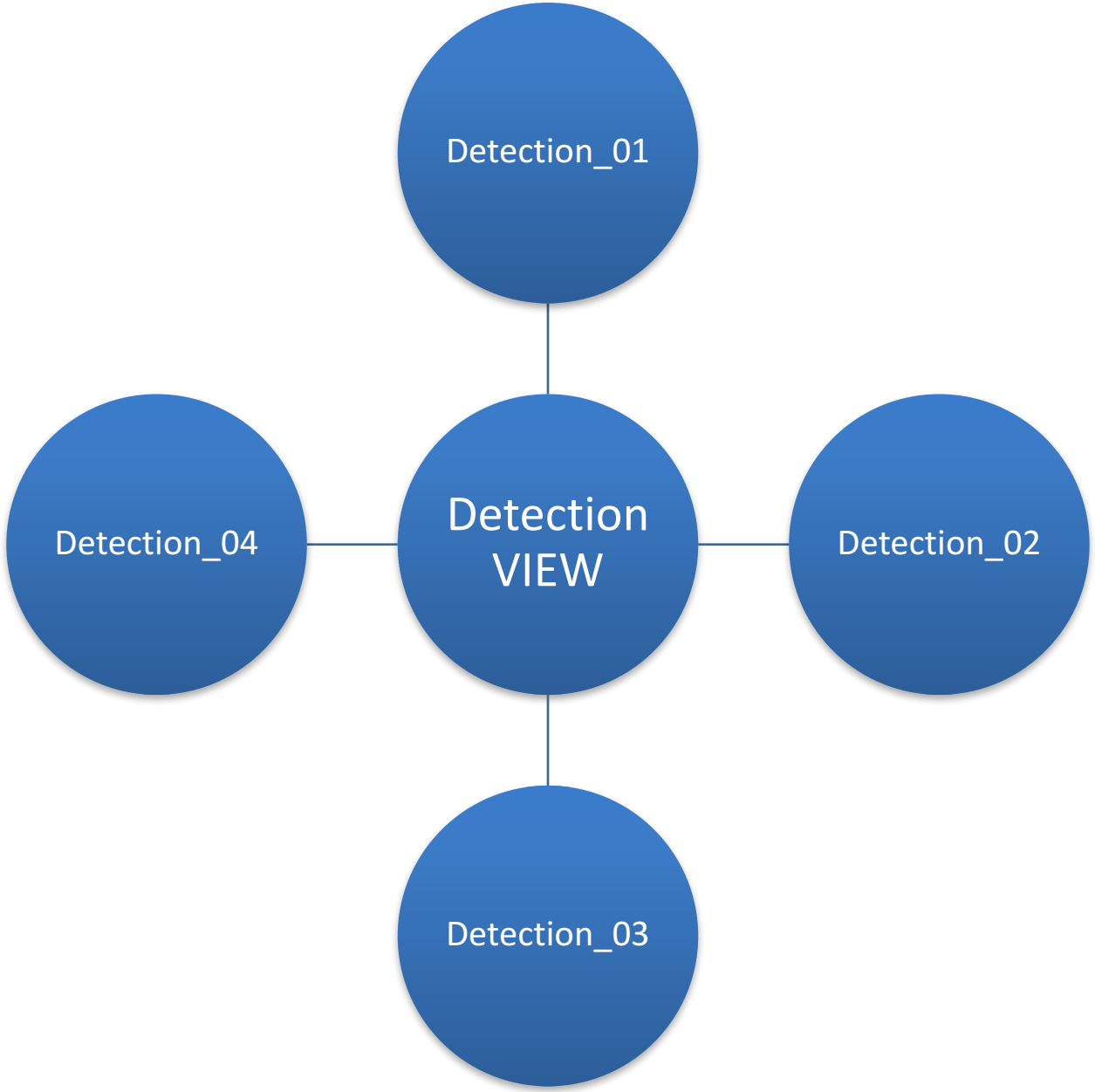
Distributed Partition Views

Detection view is created in Main DB

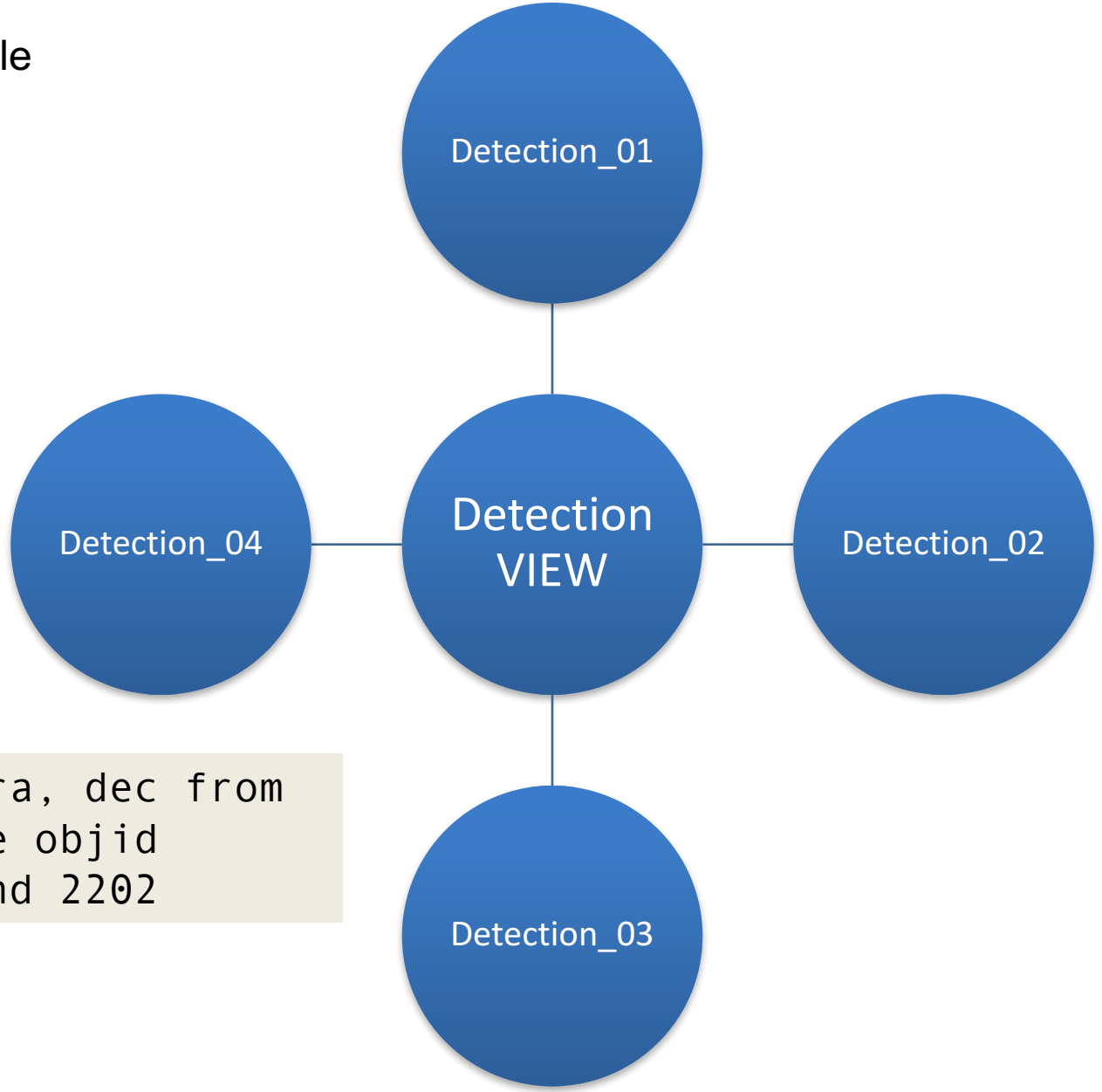
```
USE MAINDB;
CREATE VIEW Detection AS (
        SELECT * FROM Slice01.Detection_01 where objid between 0 and 999
        UNION ALL
        SELECT * FROM Slice02.Detection_02 where objid between 1000 and 1999
        UNION ALL
        SELECT * FROM Slice03.Detection_03 where objid between 2000 and 2999
        UNION ALL
        SELECT * FROM Slice04.Detection_04 where objid between 2999 and 3000
)
```

The WHERE clauses are not entirely necessary, but can help the query optimizer eliminate partitions from the scan based on objid range


Slice 01
Linked server connection
Linked server connection
Linked server connection
Slice 04
Main DB
Slice 02
Linked server connection
Linked server connection
Slice 03
Linked server connection

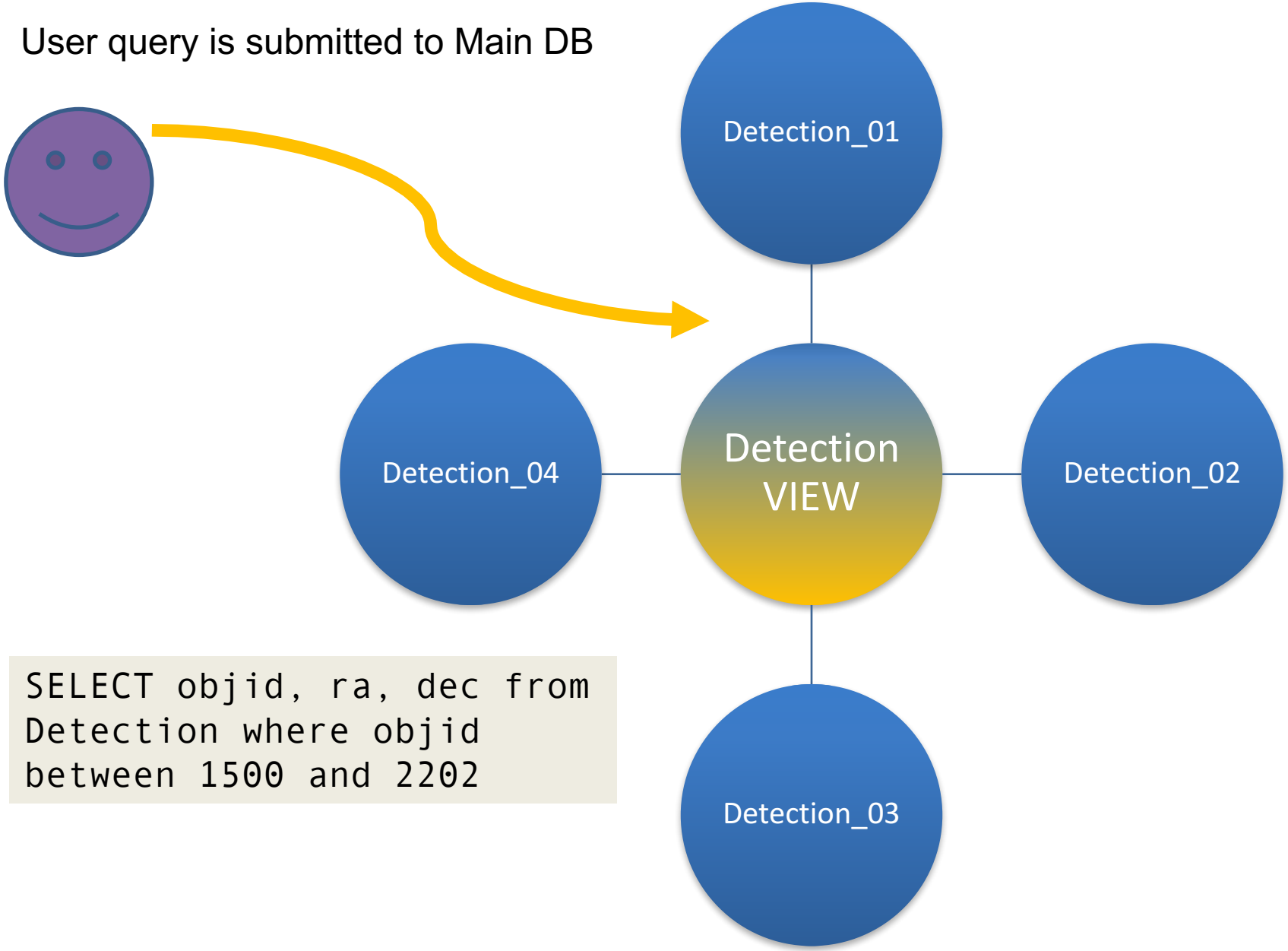

Detection_01
Detection_04
Detection VIEW
Detection_02
Detection_03

User query example




```
SELECT objid, ra, dec from
Detection where objid
between 1500 and 2202
```

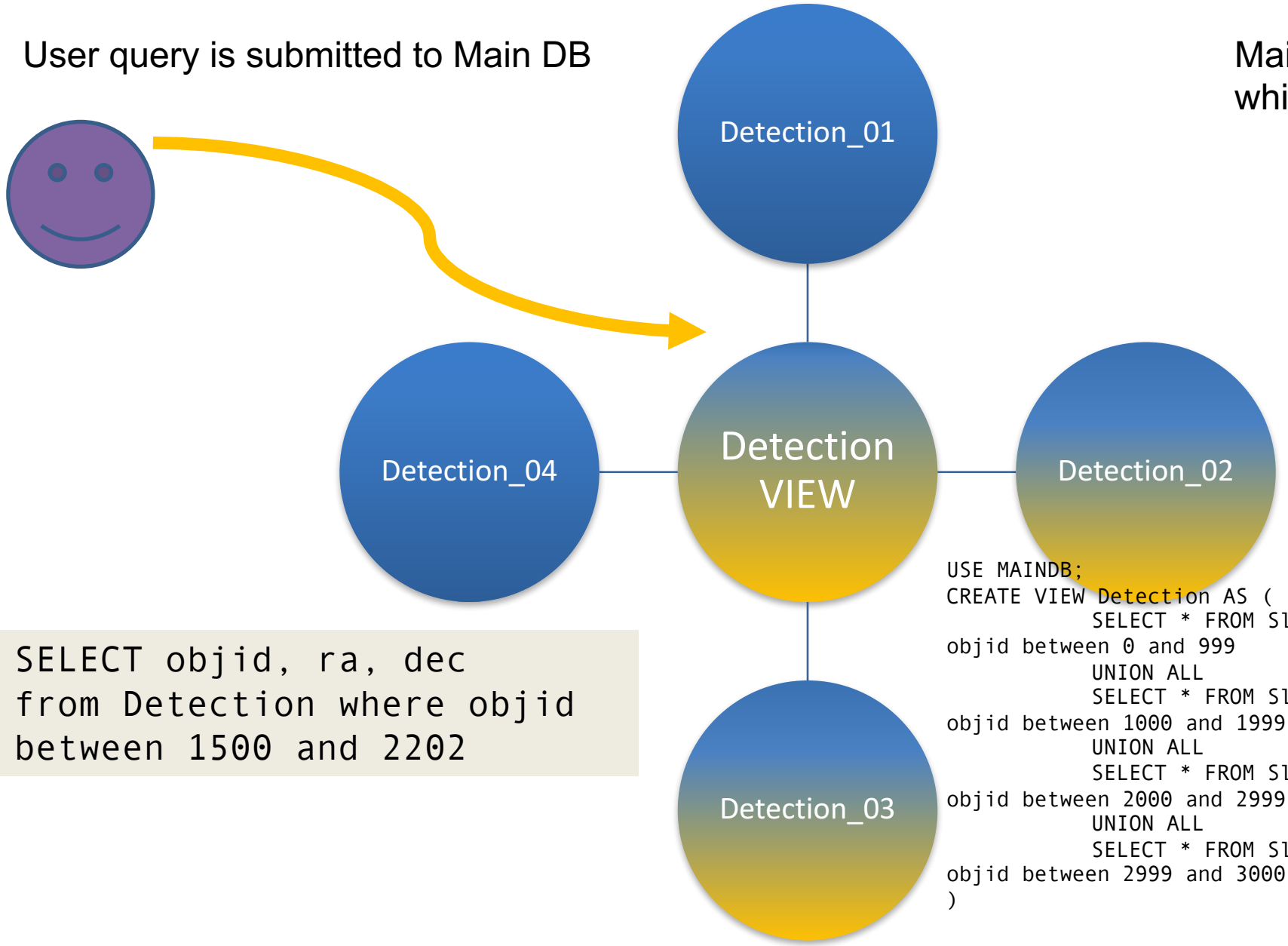
User query is submitted to Main DB




```
SELECT objid, ra, dec from
Detection where objid
between 1500 and 2202
```

User query is submitted to Main DB

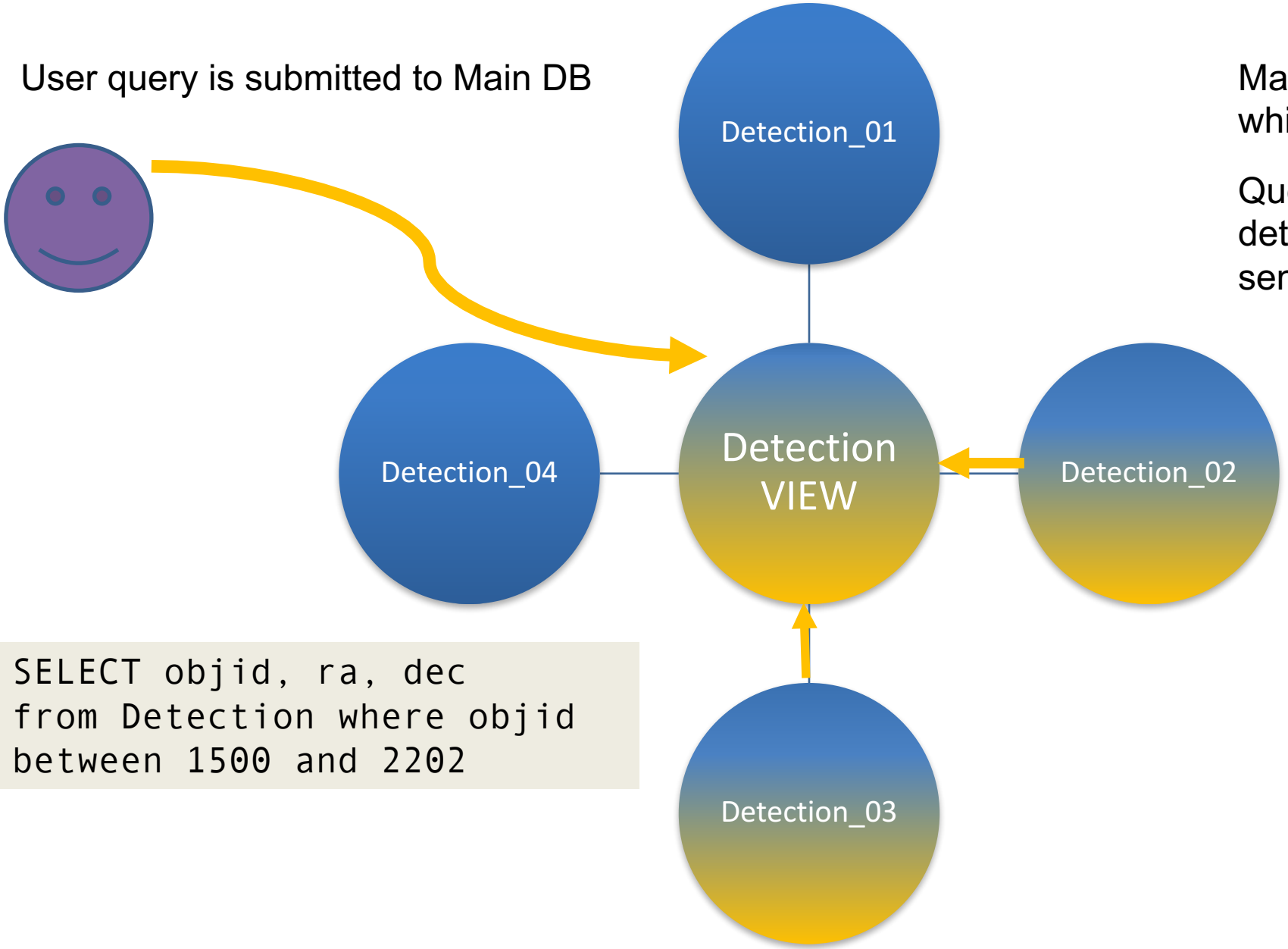
Main DB figures out which slices to query




```
SELECT objid, ra, dec
from Detection where objid
between 1500 and 2202
```

```
USE MAINDB;
CREATE VIEW Detection AS (
            SELECT * FROM Slice01.Detection_01 where
objid between 0 and 999
            UNION ALL
            SELECT * FROM Slice02.Detection_02 where
objid between 1000 and 1999
            UNION ALL
            SELECT * FROM Slice03.Detection_03 where
objid between 2000 and 2999
            UNION ALL
            SELECT * FROM Slice04.Detection_04 where
objid between 2999 and 3000
)
```

User query is submitted to Main DB

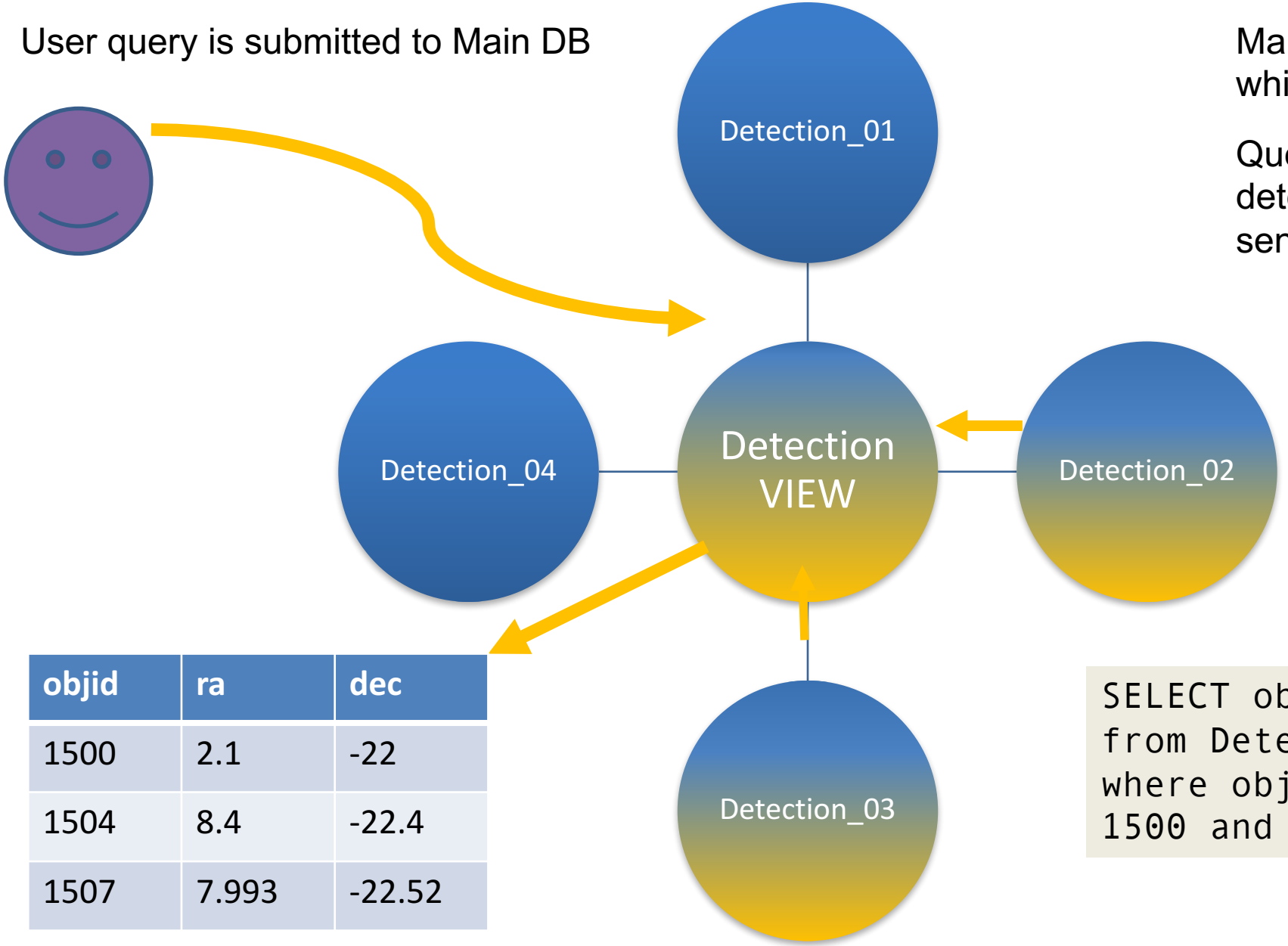


Main DB figures out which slices to query

Query is run on slice detection tables, results sent back to Main DB

```
SELECT objid, ra, dec
from Detection where objid
between 1500 and 2202
```

User query is submitted to Main DB



Main DB figures out which slices to query

Query is run on slice detection tables, results sent back to Main DB

Results sent back to user!

objid	ra	dec
1500	2.1	-22
1504	8.4	-22.4
1507	7.993	-22.52

```
SELECT objid, ra, dec
from Detection
where objid between
1500 and 2202
```

Lessons Learned from Pan-STARRS

- Distributed databases are REALLY HARD to "roll your own"
  - If you can find a native distributed DBMS, go with that!
  - You'll need a distributed development sandbox too
- Weekly load/merge was a killer
  - Detection and StackDetection tables were huge (20 G-rows)
  - We made mistakes also in how we designed "merge"
  - Ultimately this was changed to once per month, but still …
- Schema kept changing even after surveys started
  - Management didn't know how to say NO to consortium users
  - Canonical "star" schema did not serve us well in this case
- Unknown unknowns – things you couldn't predict even with good planning
  - We ran into technical limits we didn't know existed
    - Hitting the max # of dbs in one instance of SQL Server (32k)
  - New use cases were discovered once data was online
    - You cannot realistically restrict what users can do

Pan-STARRS Today

- Hosted at MAST/STScI
- The database is no longer distributed
  - There are still 32 slices + Object DB
  - But they are all on one “fat” server
  - Basic design still the same, but no need to use DPVs
- STScI has their own instance of CasJobs
  - Used for GALEX and Pan-STARRS, among others

Looking forward to LSST

- Data volume even bigger than Pan-STARRS
  - One SDSS every 2-3 nights
  - Daily load/merge, alert generation
- Which lessons are most relevant to LSST?
  - What are the good practices that it should continue?
- What should LSST do different?
  - Should it stick to relational database technology?
  - Should it put its data in the cloud?
  - What new technologies could be game changers?