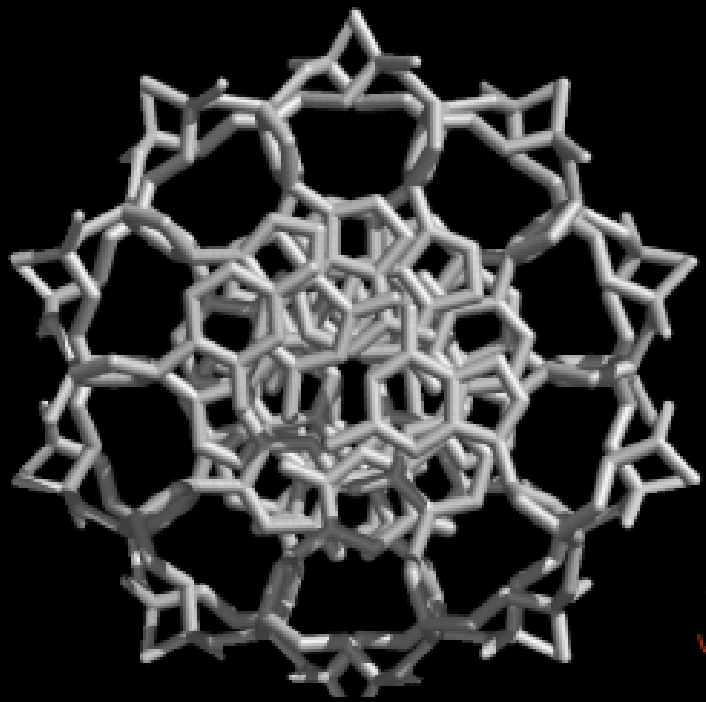


# RBVI

Resource for Biocomputing, Visualization, and Informatics

# *DASH*

## DAta SHaring Infrastructure

Resource for
Biocomputing,
Visualization, and
Informatics

# Outline

- Problem Statement
- Event Model
- DASH Description

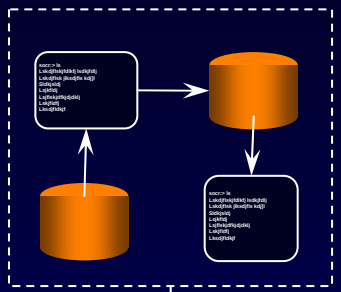
# Problem Statement

- Biological Science is increasingly a *collaborative* science
- Much of this collaboration involves the dissemination and sharing of data between disparate groups with disparate disciplines
  - Often small to medium sized labs
  - Often very different computing environments and levels of IT expertise
- Traditionally, this data sharing has been labor-intensive and error-prone
  - Data conversion issues
  - Need for automation

# Problem Statement

- Collaboration is critical, but so are the issues of data ownership and integrity
  - Desire to protect data until its “ready”
- Data processing is often necessarily ad-hoc
  - New results might require different processing approaches
  - Different researchers might process results differently for different purposes
- Existing approaches to this problem not sufficient
  - Filesharing approaches require compatible computing environments
  - Workflow approaches require significant investment in process design
  - Grid approaches require compatible computing environments and a level of shared security infrastructure

# Example: Baygenomics



## Genomics Core Facility

Gene trapping experiments
Sequencing

Sequence data
40 – 500 bp

Trim sequence data

Reject bad sequences

Refined sequence data

Blast against GenBank NR

Populate database

Confirmation of id

Putative gene id for trap

Create .html files

HTML

HTML

Update web page

Blast against genomic data

Create .png image

PNG

# Solution: The Event Model

**Event model is based on responses to *events***

- **Events could include file updates, database record updates, e-mail delivery, web service messages, web posts, time**

**The response to an event depends on an *event handler***

- **Event handlers are programs designed to deal with a specific event**
- **Can have multiple event handlers for a single event**

**Events can be combined logically**

- **Boolean operations of AND, OR, and NOT, can be used to create “virtual events” to trigger specific event handlers**

# DASH's Event Model

DASH utilizes the event model

Currently supported event types:

- File update
  - E-Mail is handled through file update
- Database record update

Currently supported Boolean combinations:

- OR, AND, NOT

Event Handlers:

- Similar to UNIX filters
- Take an input file and transform it to an output file
- Output file can be used to trigger additional event handlers
  - Can be explicit or implicit
- Security is handled utilizing native security (file system or database)

Event registration

- DASH events and event handlers are registered in a MySQL Database

# Why Use an Event Model ?

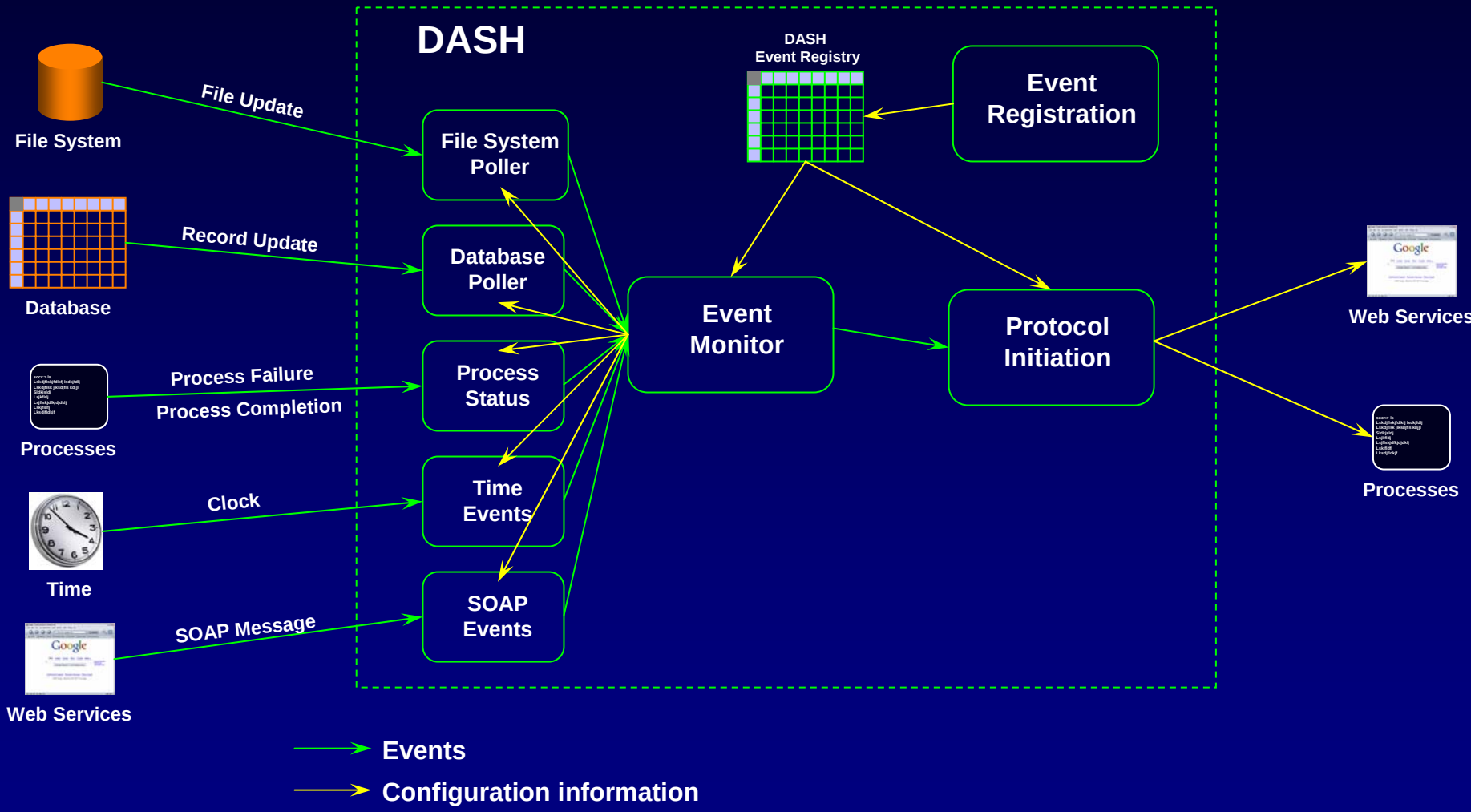
## Advantages

- Multiple events can be combined sequentially to create a data flow
- Events are fine-grained, so lend themselves to ad-hoc definition and re-definition
- Events don’t depend on underlying infrastructure
  - Can utilize data sharing technologies, database technologies, Web Services, E-Mail or FTP to trigger an event
  - Provides for a multitude of underlying mechanisms
- Event approach does not assume any specific shared security realm
  - Triggering an event does not imply any global security
  - Security for objects (files, records, etc.) can be fine-grained

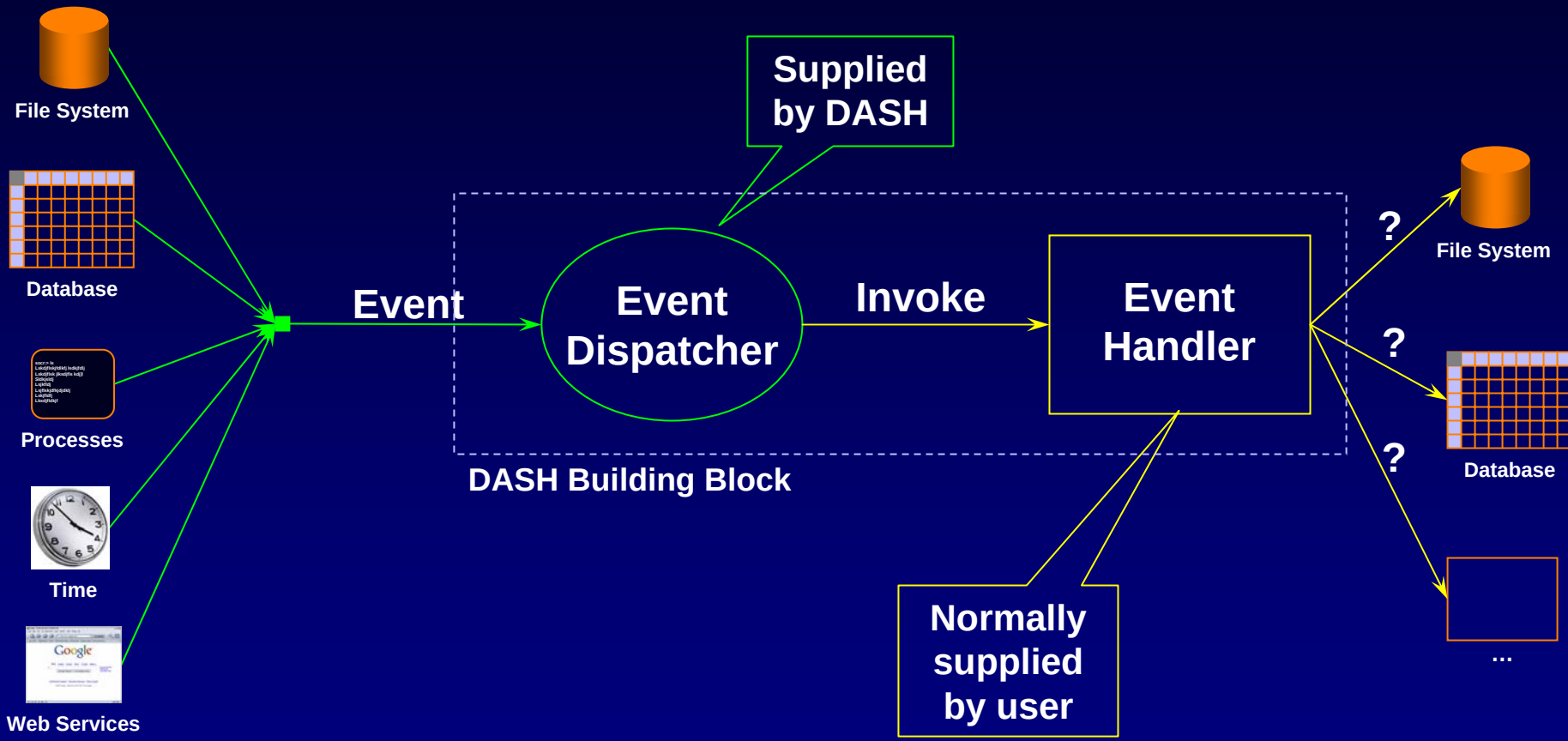
## Disadvantages

- Does not explicitly deal with data delivery
- Does not provide for process-centric validation and control
  - Not appropriate for business

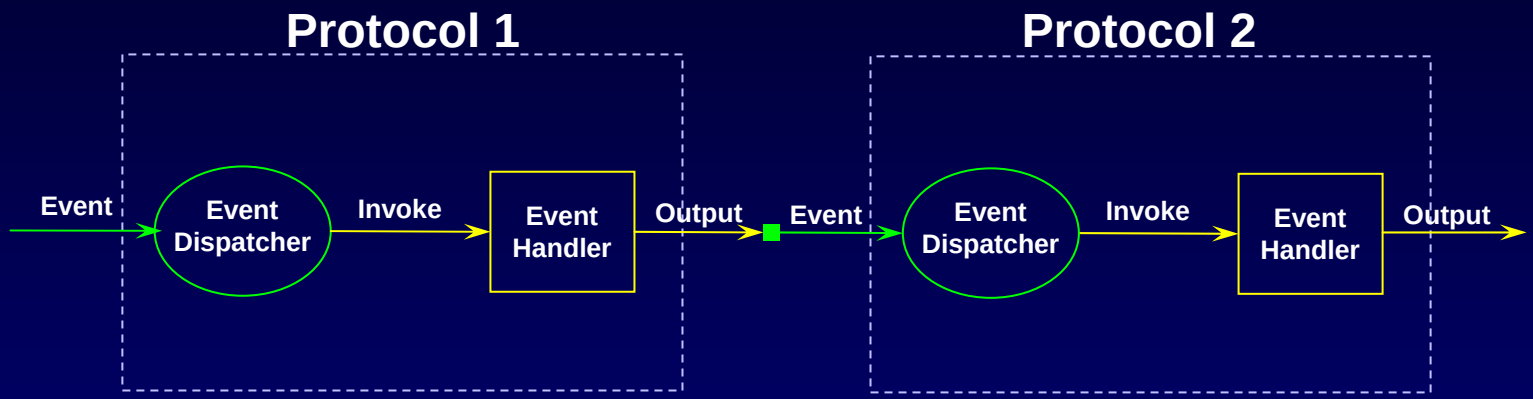
# DASH Conceptual Architecture

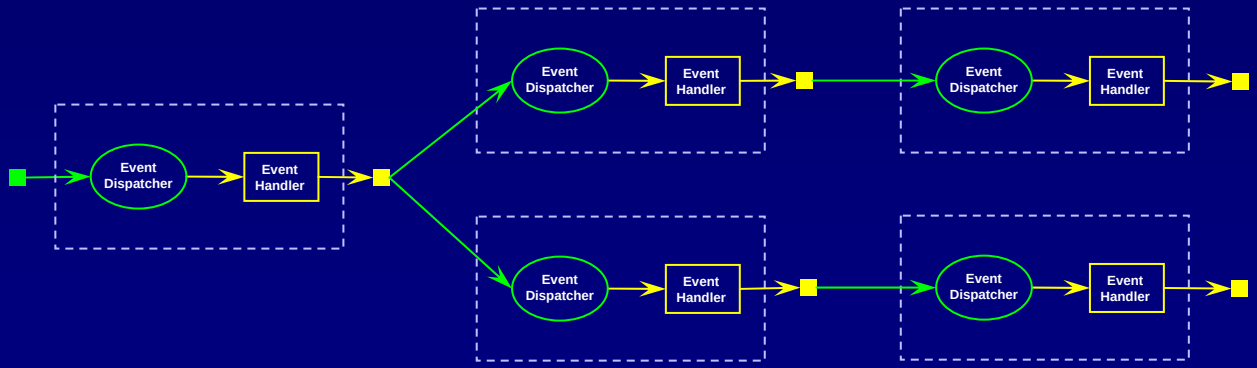


# DASH Event Model



# DASH Event Model – Pipelines

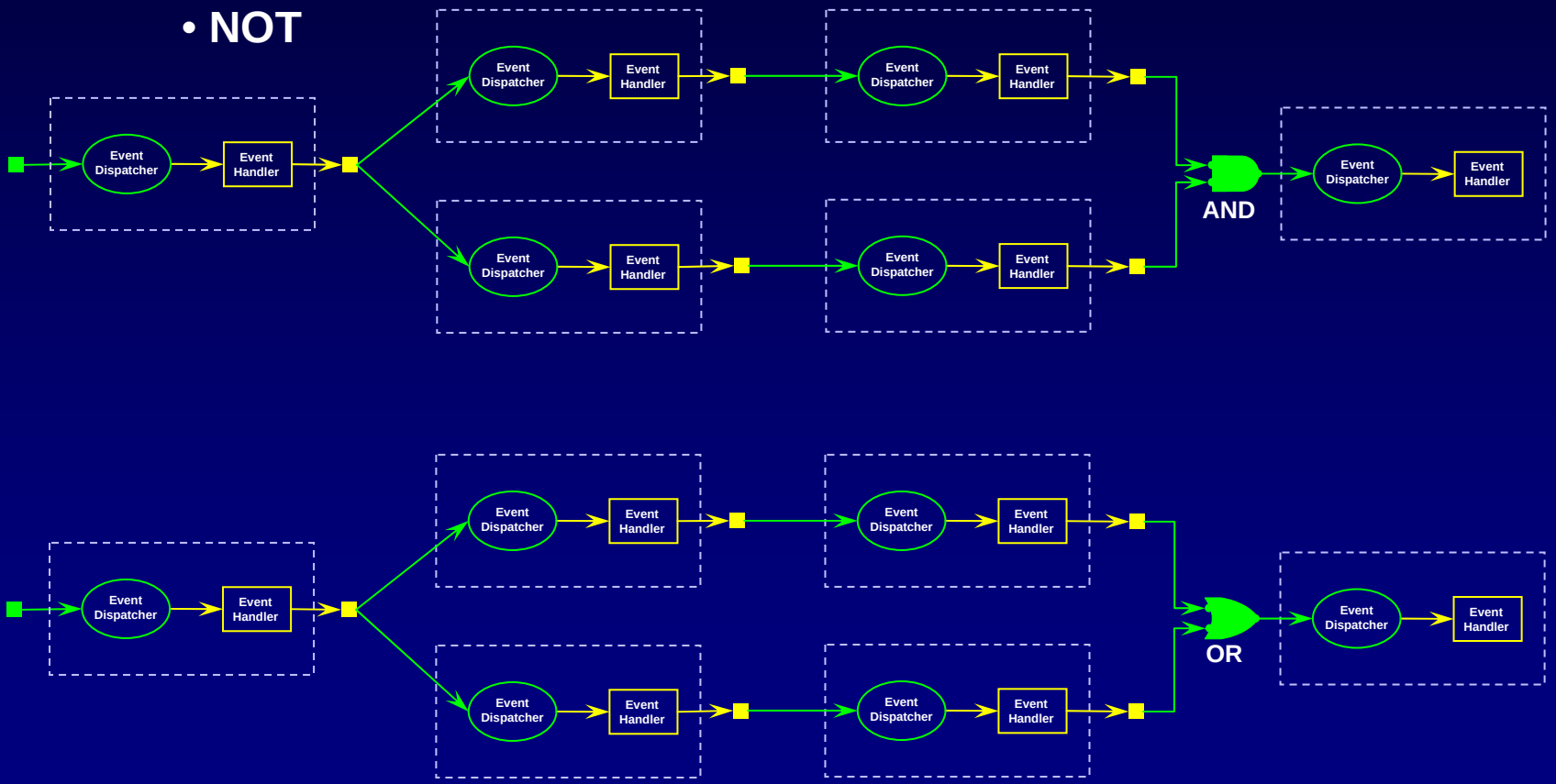


# DASH Event Model – Virtual Events

**Events can be combined using Boolean operations:**

- **AND**
- **OR**
- **NOT**

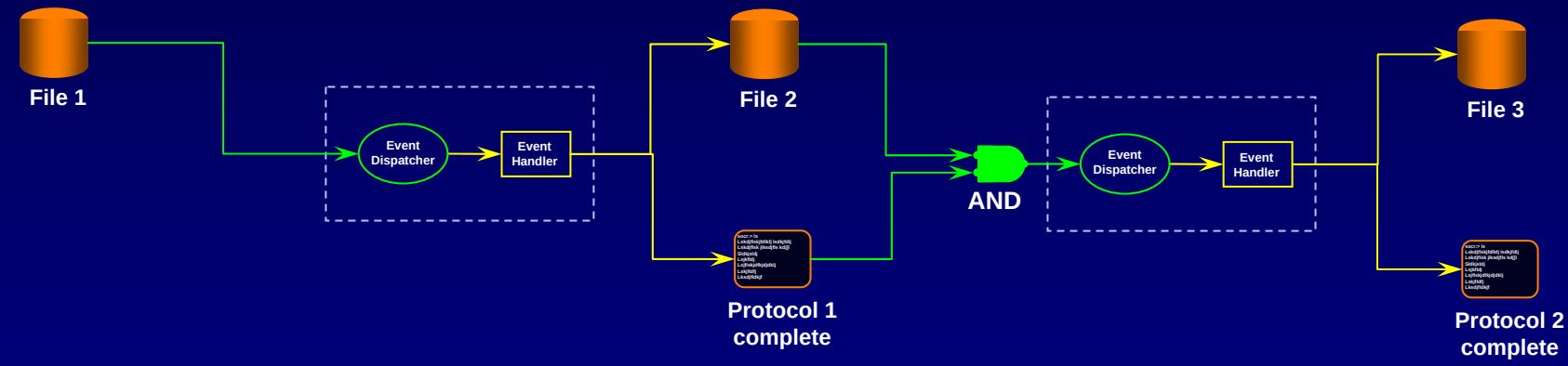


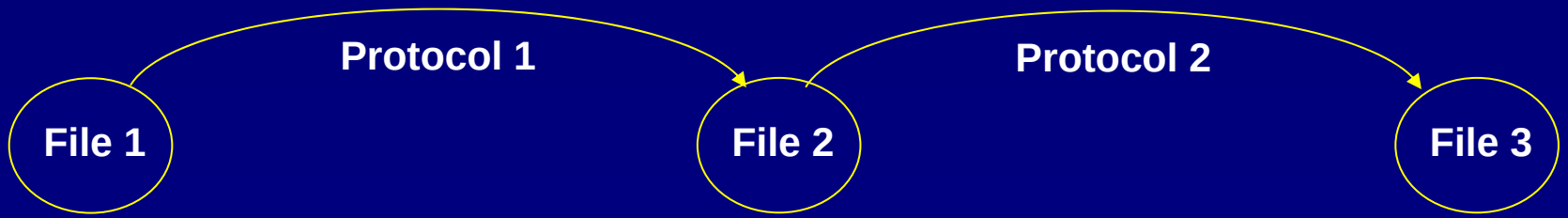
# DASH Event Model – Patterns

**One standard event pattern:**

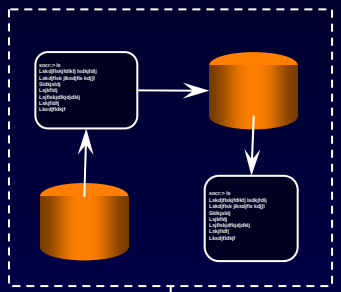
- **IF a file has been updated,**
- **AND a certain process has completed**
- **THEN start the next process**

**Similar to steps in a Data Flow Diagram**



# A Closer Look



## Genomics Core Facility

Gene trapping experiments
Sequencing

Confirmation of id

Blast against
GenBank NR

Trim sequence
data

Populate
database

Putative gene id
for trap

Update web
page

Create .html
files

Sequence data
40 – 500 bp

Refined sequence
data

Reject bad sequences

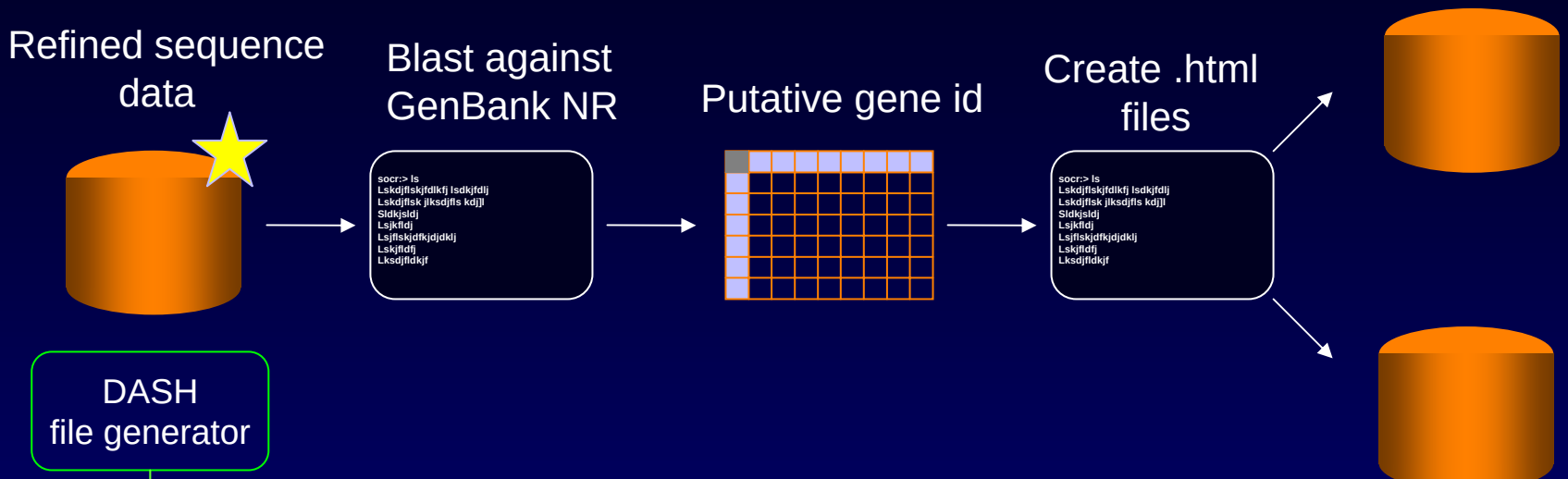
Blast against
genomic data

Create .png
image

HTML

HTML

PNG

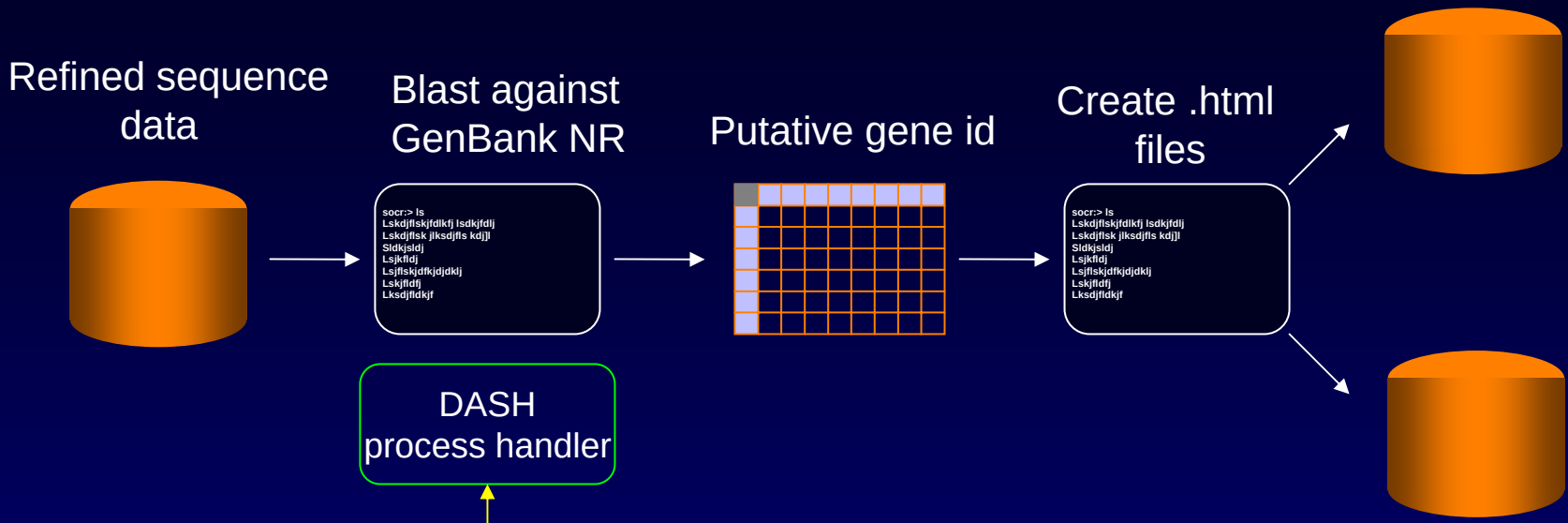



**file modification event**

## DASH Event Dispatcher

event

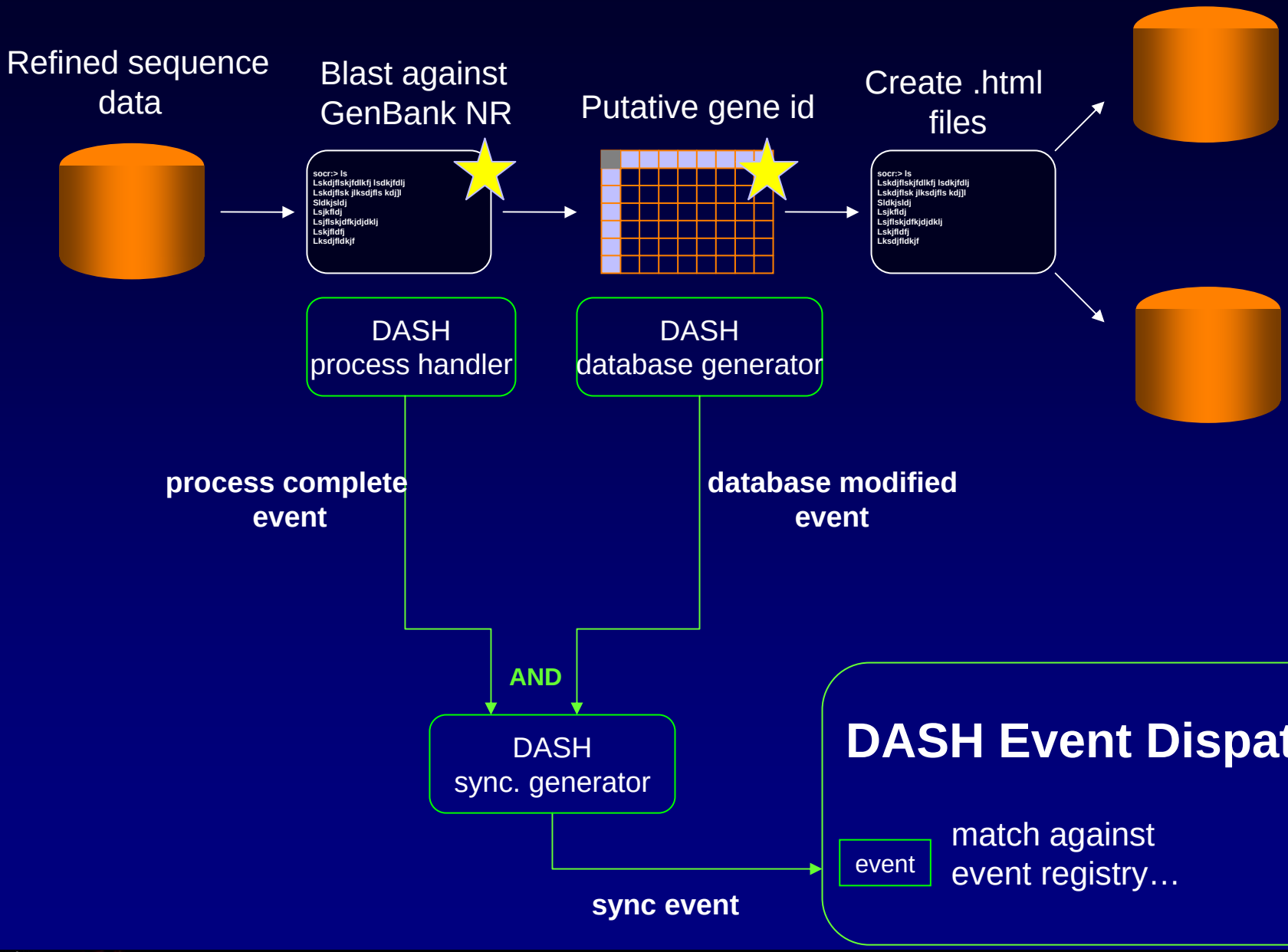
match against event registry…..

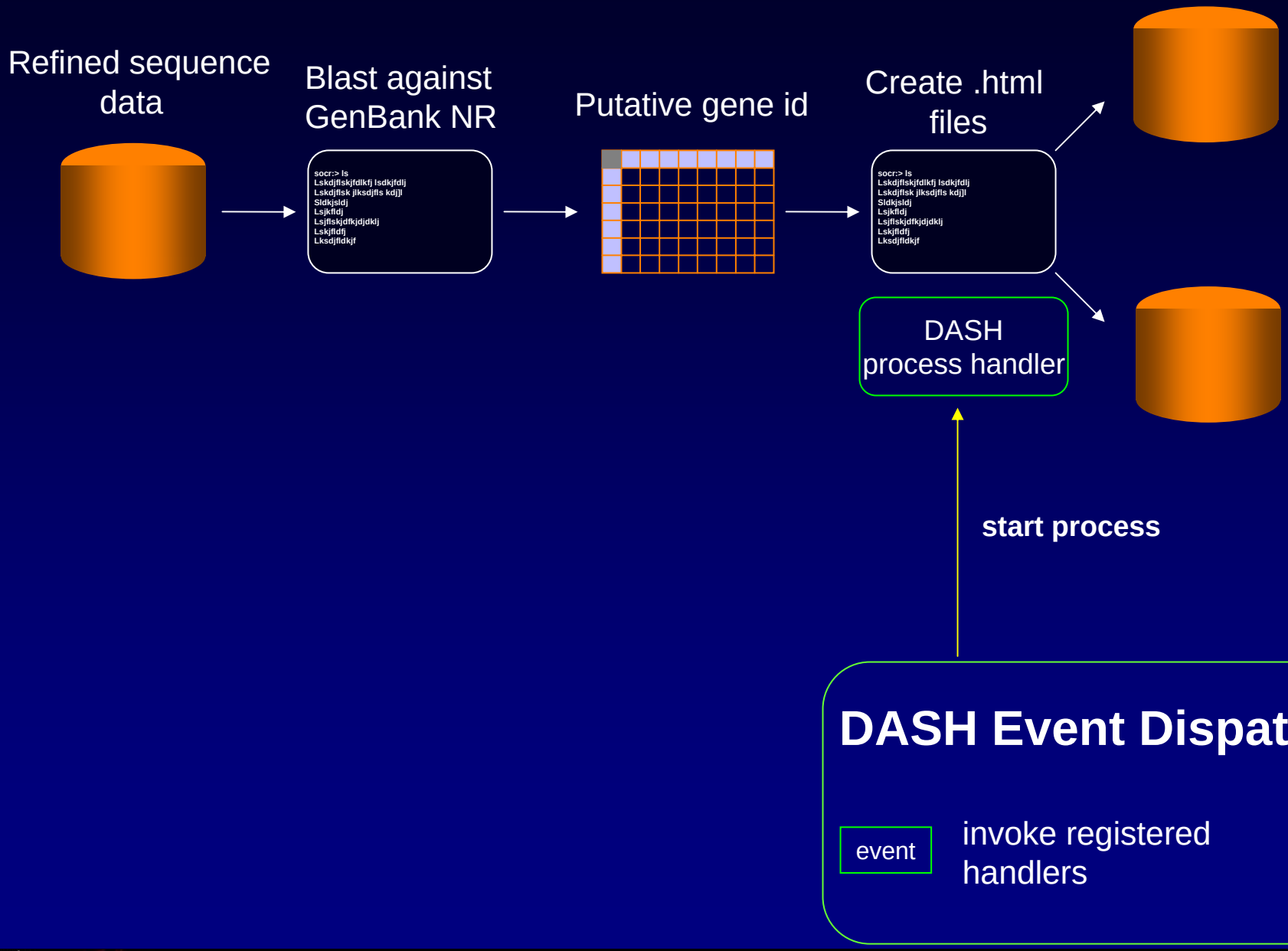


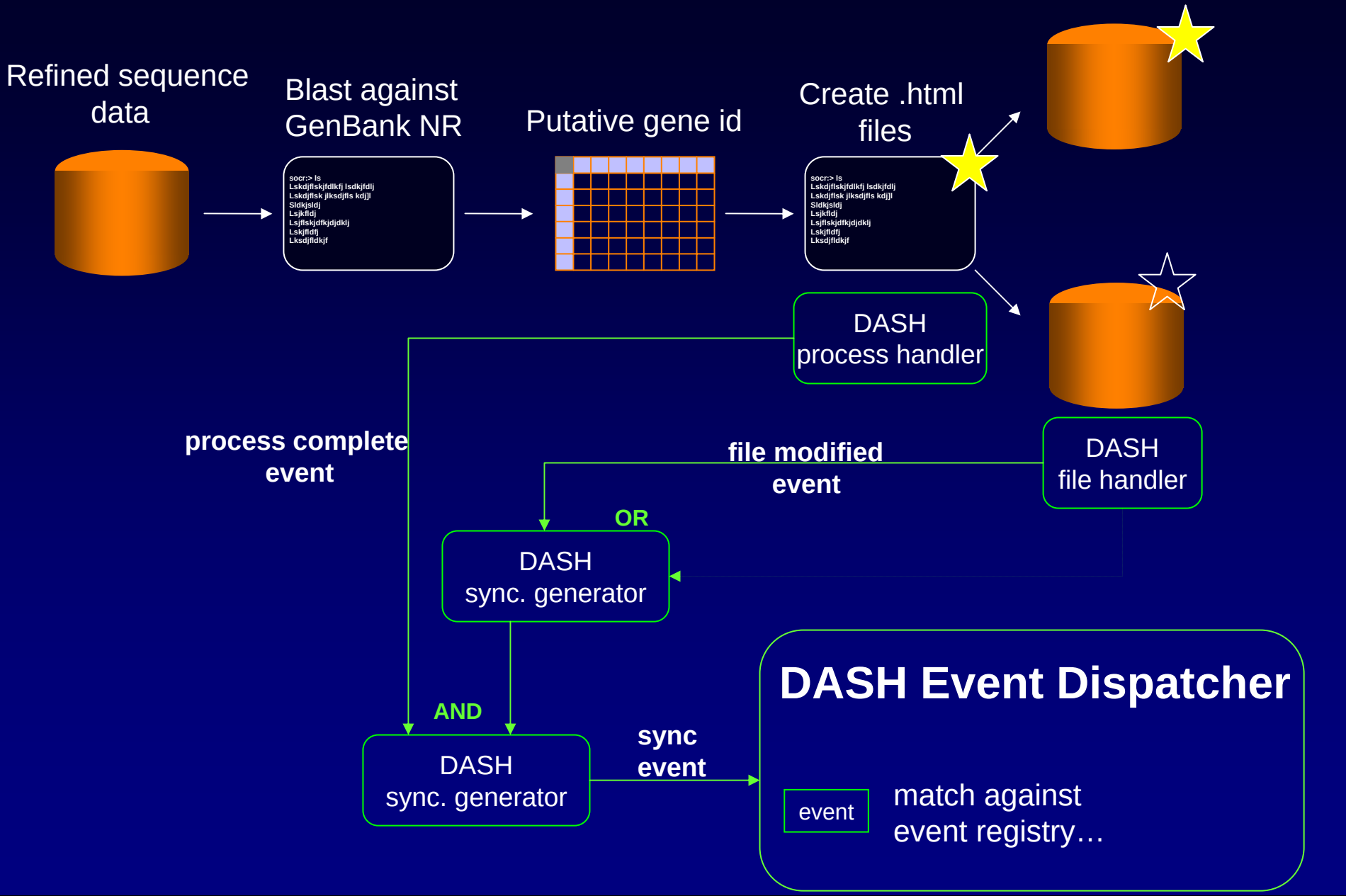

## DASH Event Dispatcher

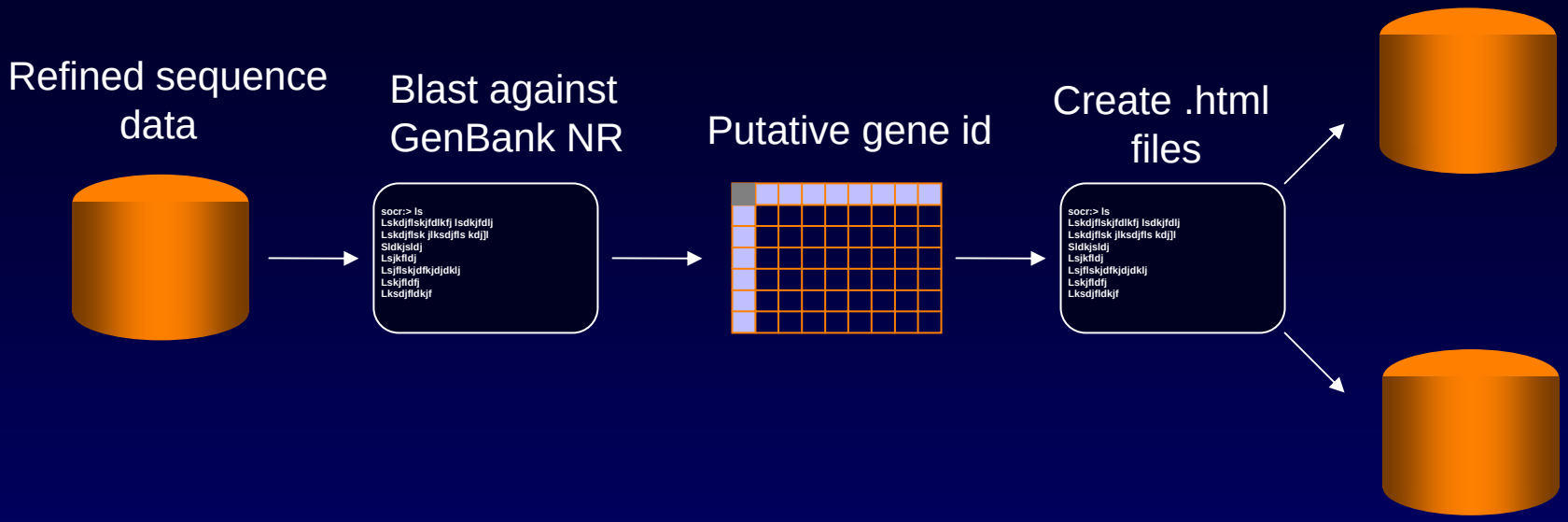
event

invoke registered handlers


Refined sequence data
Blast against GenBank NR
Putative gene id
Create .html files
socr:> ls
Lskdjflskjfdlkfj lsdkjfdlj
Lskdjflsk jlksdjfls kdjjl
Sldkjsldj
Lsjkfldj
Lsjflskjdfkjdjdklj
Lskjfldfj
Lksdjfldkjf
DASH process handler
DASH database generator
process complete event
database modified event
AND
DASH sync. generator
sync event
DASH Event Dispatcher
event
match against event registry…


Refined sequence data
Blast against GenBank NR
Putative gene id
Create .html files
socr:> ls
Lskdjflskjfdlkfj lsdkjfdlj
Lskdjflsk jlksdjfls kdjjl
Sldkjsldj
Lsjkfldj
Lsjflskjdfkjdjdklj
Lskjfldfj
Lksdjfldkjf
DASH process handler
start process
DASH Event Dispatcher
event
invoke registered handlers


Refined sequence data
Blast against GenBank NR
Putative gene id
Create .html files
socr:> ls
Lskdjflskjfdlkfj lsdkjfdlj
Lskdjflsk jlksdjfls kdjjl
Sldkjsldj
Lsjkfldj
Lsjflskjdfkjdjdklj
Lskjfldfj
Lksdjfldkjf
DASH process handler
DASH file handler
process complete event
file modified event
OR
DASH sync. generator
AND
DASH sync. generator
sync event
DASH Event Dispatcher
event
match against event registry…




???

## DASH Event Dispatcher

event

invoke registered handlers

# DASH Futures

New Event Types:

- Web Service (SOAP) messages
- Web CGI posts

DASH protocol documentation

- Export of DASH protocols using Web Services Choreography Description Language
- Discovery of implicit protocols using Web Services

Ease-of-use improvements

- Web-based registration of events and handlers
- Web-based monitoring of events and handler progress
- Provided wrappers for common applications
- “Generic” wrapper to quickly wrap commercial applications
  - Requires a command-line interface

Extensibility

- Use of WSDL and SOAP to extend DASH events across a network

# Questions ?

# Workflow Model

## Workflow systems utilize a process-centric model

- Significant investment in process design
- Processes explicitly involve external steps (e.g. approval)
- Most implementations are very business-centric

## Workflow systems are very appropriate for business

- Processes are well-defined and stable
- Need for documentation and agreement of process implementations
- Up-front investment in process design has significant payback

## Example systems:

- BEA, Taverna, TIBCO

# Workflow Model

## Advantages

- Rigorous process-oriented definition
- Substantial tool support for process definition
- Well-adapted to business where processes need to be reviewed

## Disadvantages

- Definition process generally requires dedicated resources
  - Process analysts
- Inhibits ad-hoc definitions
- Integration is often very expensive
- Generally centralized

# Data Sharing Model

**Data sharing approaches provide for data *availability***

- **All data available, all the time**
- **Requires substantial agreements about formats and processing rules**
- **Assumes some level of shared security realm**
- **Requires a level of system compatibility**
- **Potential for significant management expenses**

**Example systems:**

- **Global File System (GFS), NFS, Avaki, Lustre, Centralized Databases**

# Data Sharing Model

## Advantages

- Provides for globally shared data
- Data is available “instantaneously” to all participating users
- Some systems provide for built-in data conversion (e.g. Avaki)

## Disadvantages

- Must have standardized infrastructure
  - Packages may not work on all systems
- Significant implementation overhead
- Implies shared (or at least agreed-to) security realms
- No provision for workflow

# Batch Management Systems

**Batch management systems support the queuing and management of processes**

- **Often across multiple systems (distributed)**
- **Sometimes utilizing underlying distributed libraries (computational clusters, Grid)**

**Role of a batch management system is to efficiently manage computational resources**

- **Explicit goal is to increase throughput**

**Batch management systems often include some facility for handling input and output files**

**Example systems:**

- **Platform (LSF, Symphony), NQS, PBS, Globus, SunGrid**

# Batch Management Systems

## Advantages

- Efficiently manages processes across distributed computers

## Disadvantages

- No facility to trigger off of data-driven events
- Generally requires broad integration and cooperation amongst all participants