Straxferno

Vestibule: using strax

Straxferno course

Aims:

- Train expert analysts and developers
- Spread knowledge of strax(en) internals
- Train new strax maintainers

This is a **discussion course**. There are (almost) no lectures.

- Helps to do readings before we discuss them
- Study questions: discussed together
- Exercises: individual work, could discuss if useful

Today

Intro

- Welcome!
- Why / what is strax?
- Structure of course

Questions / discussion

- Study questions
- Your questions

Preview of next week

Why strax?

100x faster processing

Full processing as online monitoring

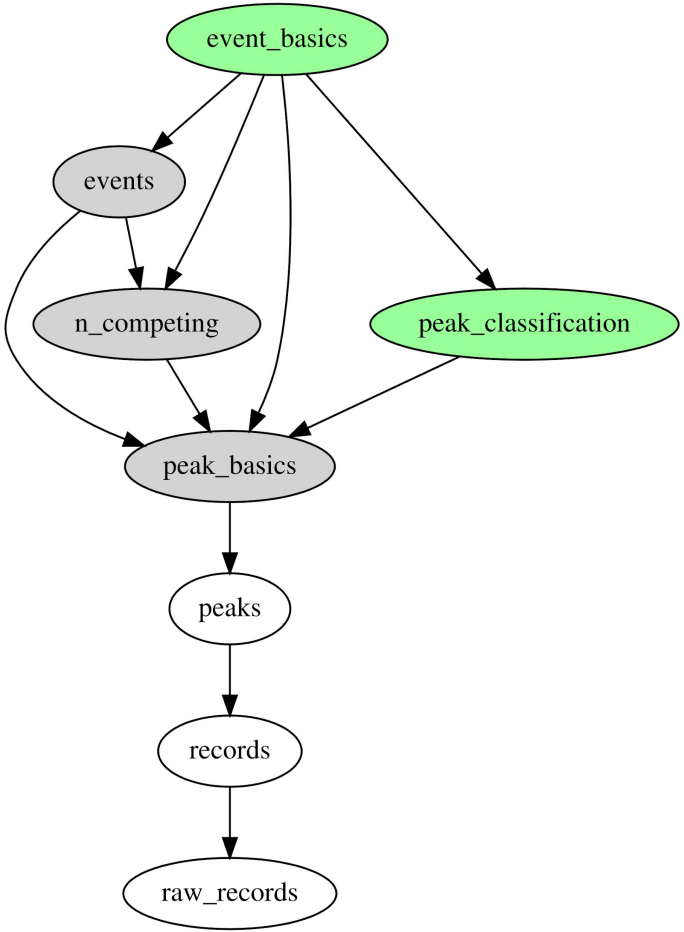
Improved analysis experience

Easy to change low-level settings

Data reduction without a trigger

Save everything while using less storage

What is strax really?



Analyst:

“Load `event_basics` for run 142,
with S1 coincidence requirement = 2”

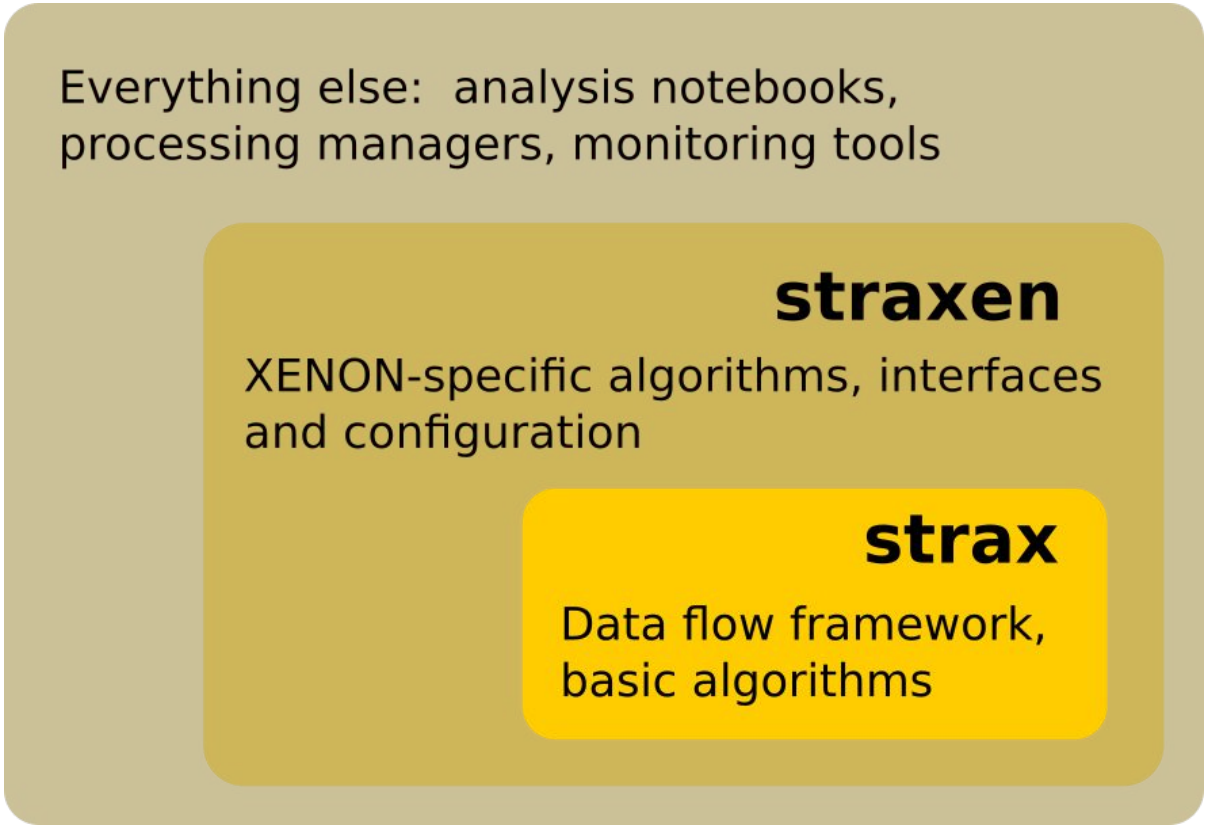
Strax, behind the scenes:

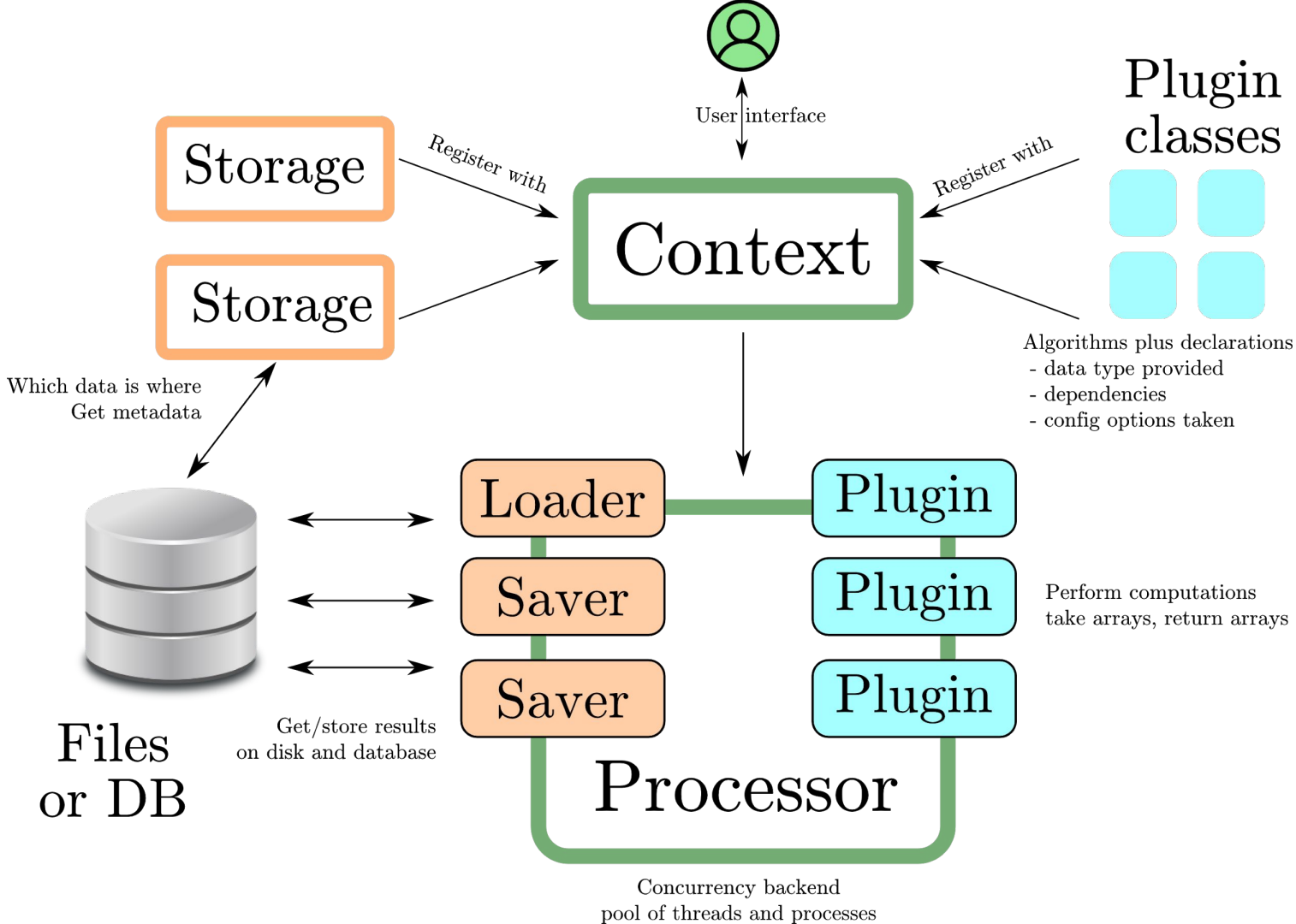
1. Build graph of dependencies
2. Find which data types can be loaded
3. Compute others that are needed

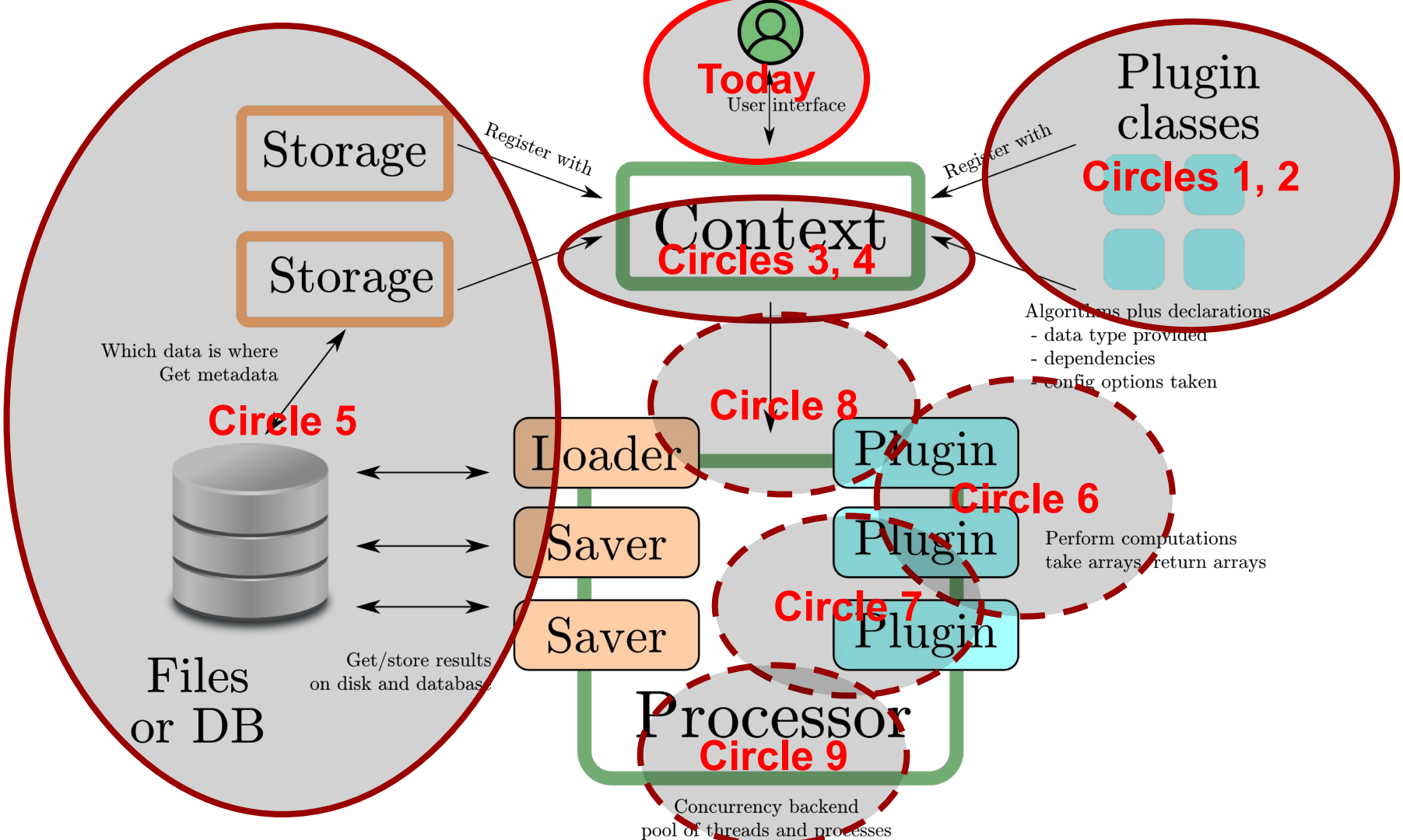
Analyst receives dataframe

Computed dataframes are now stored
(with tracking of the custom options)

The software stack




User interface
Storage
Register with
Context
Register with
Plugin classes
Storage
Algorithms plus declarations
- data type provided
- dependencies
- config options taken
Which data is where
Get metadata
Loader
Plugin
Saver
Plugin
Perform computations
take arrays, return arrays
Saver
Plugin
Files or DB
Get/store results
on disk and database
Processor
Concurrency backend
pool of threads and processes


Today
User interface
Plugin classes
Circles 1, 2
Register with
Register with
Storage
Storage
Context
Circles 3, 4
Algorithms plus declarations
- data type provided
- dependencies
- config options taken
Which data is where
Get metadata
Circle 5
Circle 8
Loader
Saver
Saver
Plugin
Plugin
Plugin
Circle 6
Perform computations
take arrays, return arrays
Circle 7
Files
or DB
Get/store results
on disk and database
Processor
Circle 9
Concurrency backend
pool of threads and processes

Session	Date
0: Using strax	Friday 15 May 14:30
1: High-level processing	Tuesday 19 May 15:00
2: Low-level processing	Tuesday 26 May 15:00
3: Loading data and mini-analyses	Tuesday 2 June 15:00
4: Contexts	Tuesday 9 June 15:00
5: Storage	Tuesday 16 June 15:00
6: Plugin implementation	Tuesday 23 June 15:00
7: Data flow	Tuesday 30 June 15:00
8: Processing setup	Tuesday 7 July 15:00
9: Mailboxes	Tuesday 14 July 15:00

How **NOT** to ask for help

“Hi Jelle, my kernel is crashing, what should I do?”

“Anyone else getting a lot of ValueErrors lately?”

How to ask for help

Read the analysis guide. See also its linked resources. Check / search gitter. Search google and stackoverflow.

Post in gitter (analysis, students_only, or strax), using gist if it's long:

- What **environment** + strax(en) **versions** are you running
- What **code** are you running
- What **error + traceback** do you get
- What did you already try and did not work

If you suspect a strax(en) bug, you can also open a github issue

Got help? Pass on the good spirit and help others.