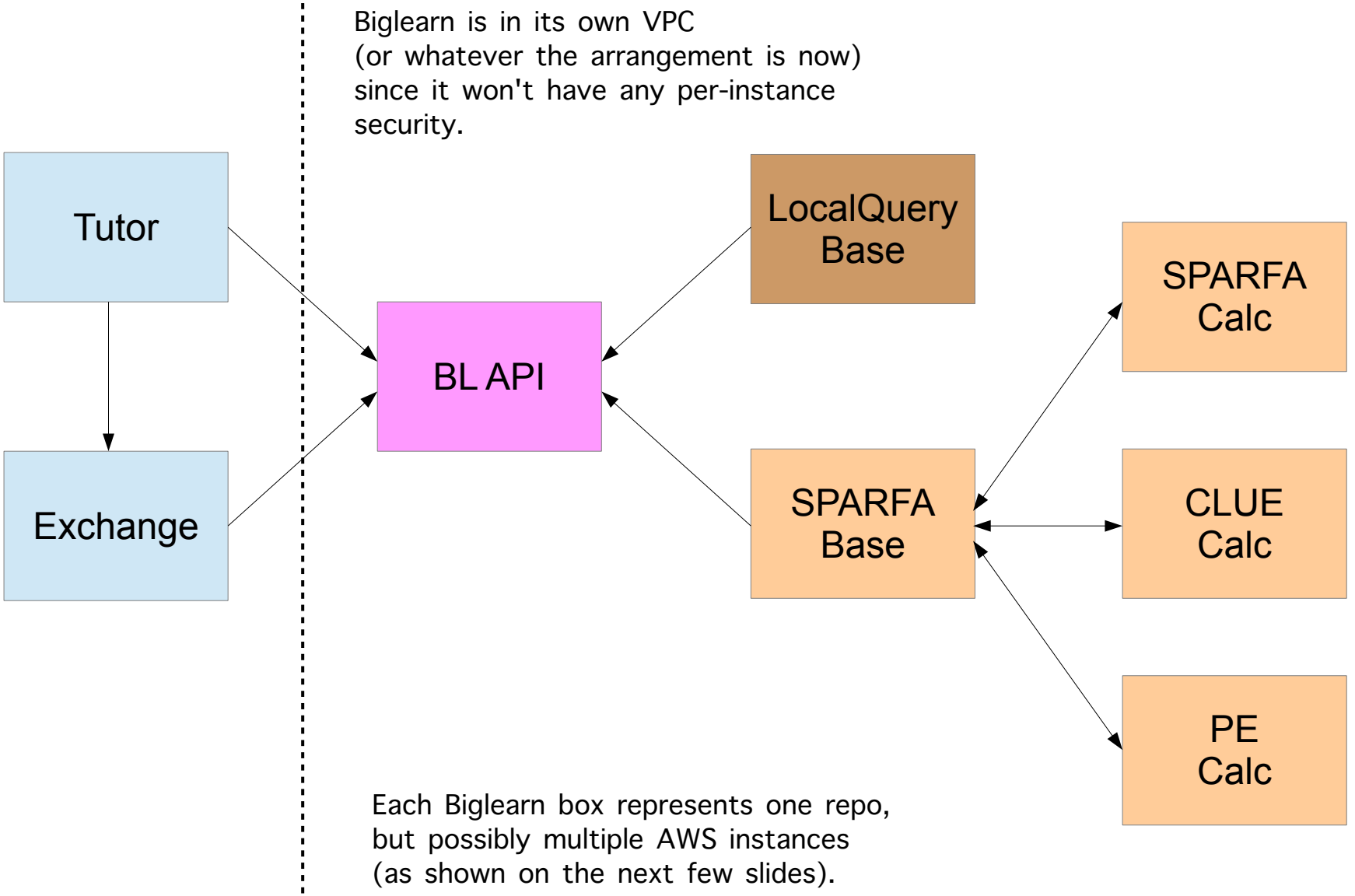
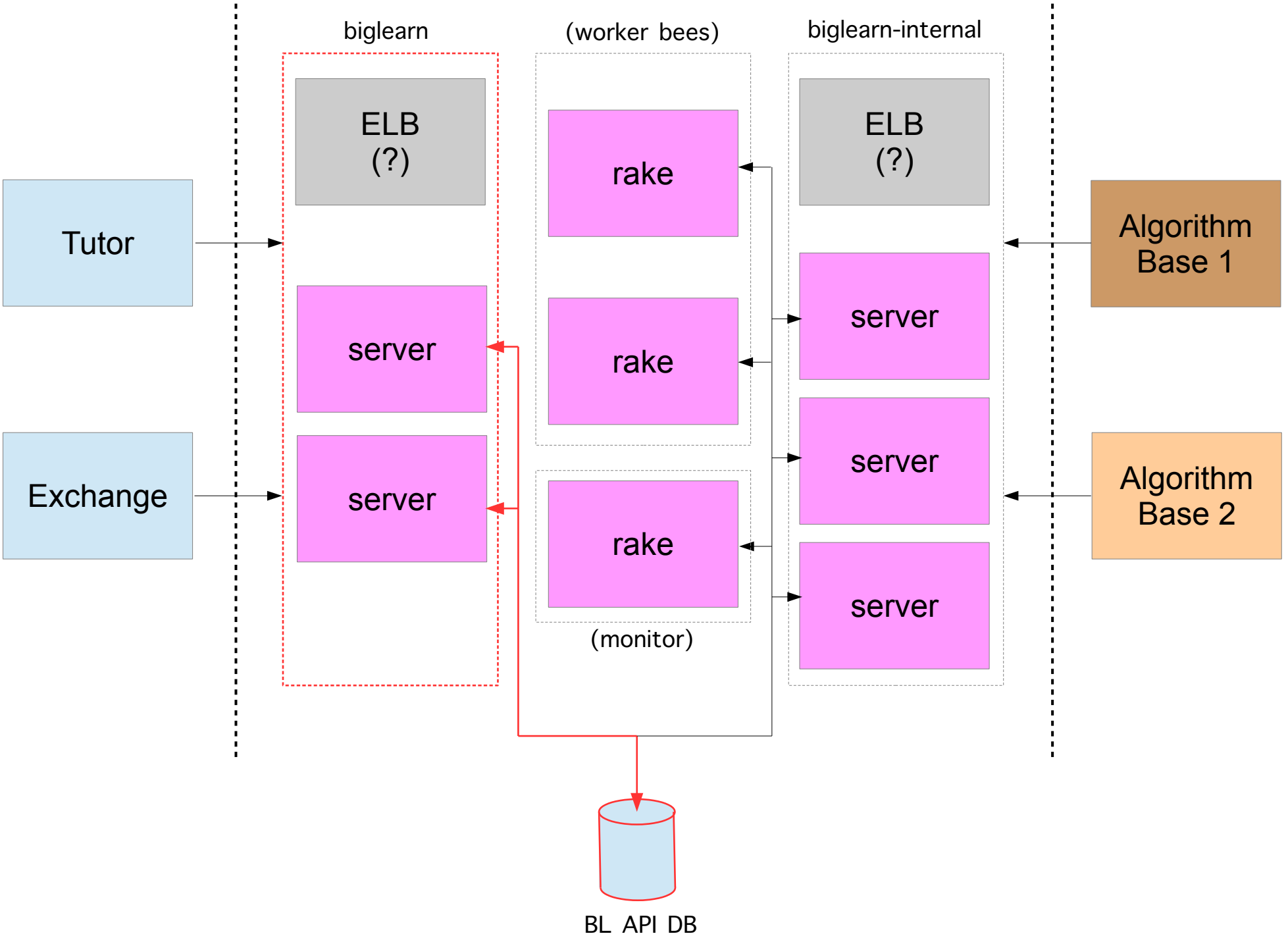
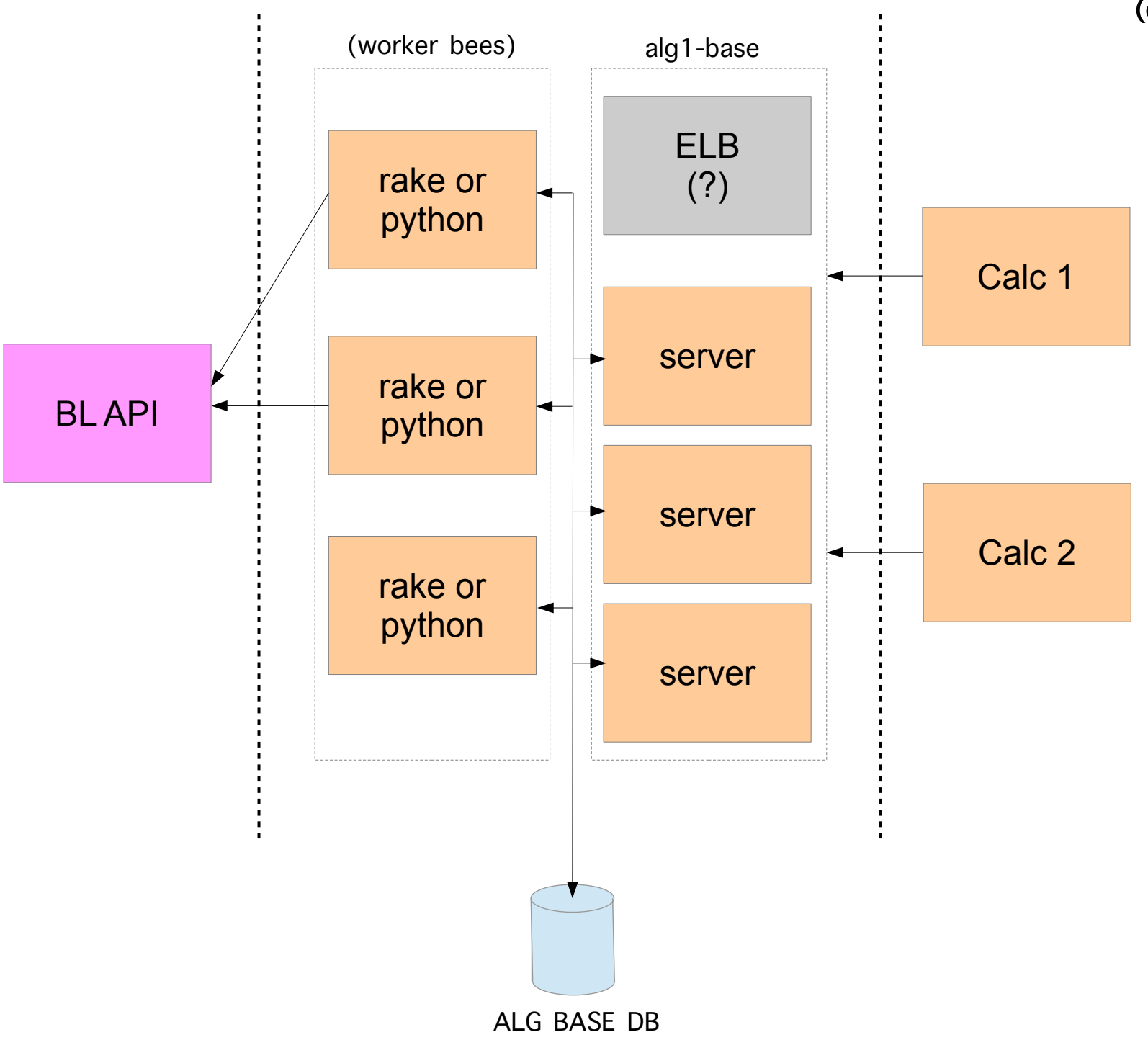

Biglearn is in its own VPC
(or whatever the arrangement is now)
since it won't have any per-instance
security.
Tutor
Exchange
BL API
LocalQuery
Base
SPARFA
Base
SPARFA
Calc
CLUE
Calc
PE
Calc
Each Biglearn box represents one repo,
but possibly multiple AWS instances
(as shown on the next few slides).

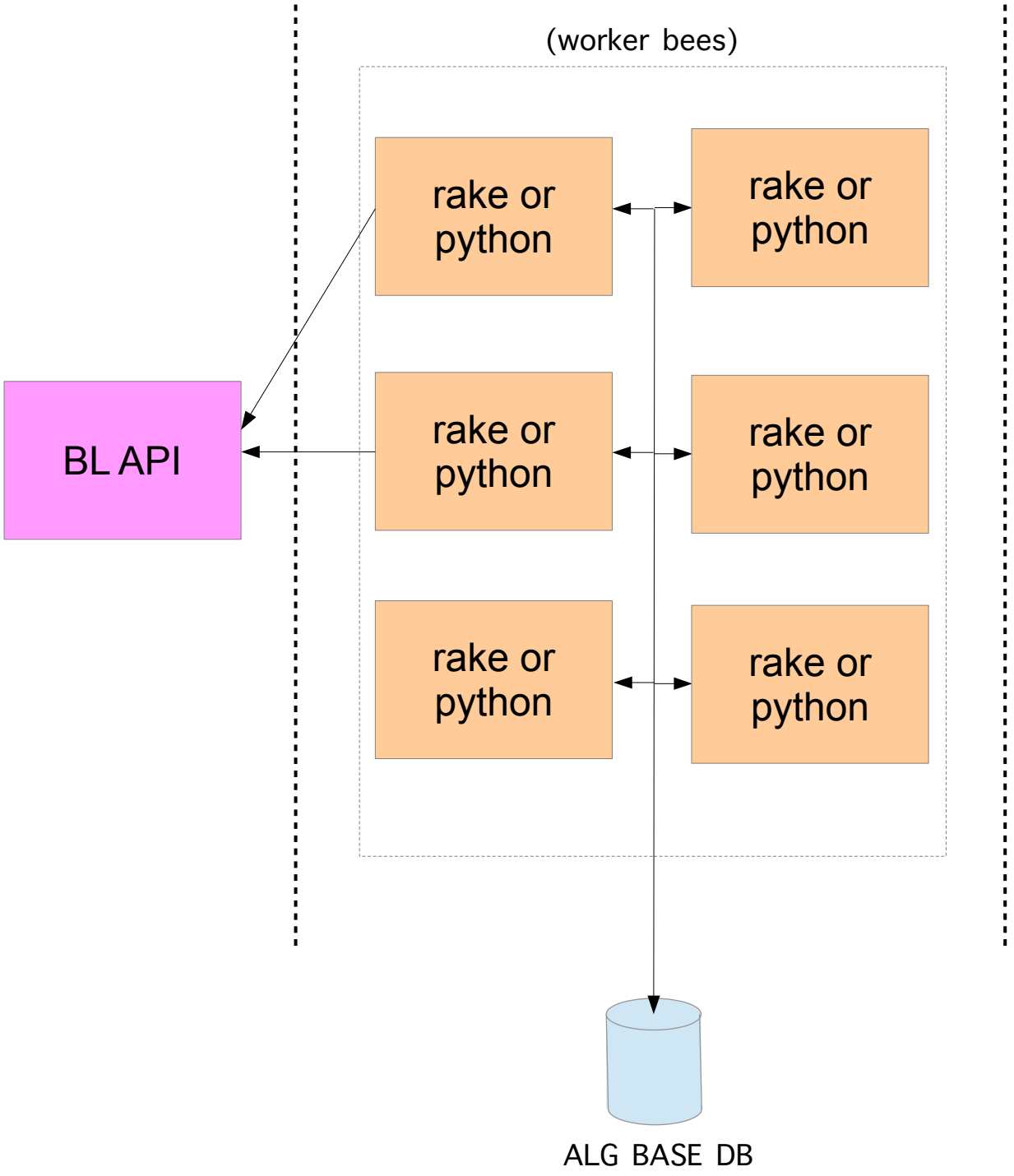
BL API

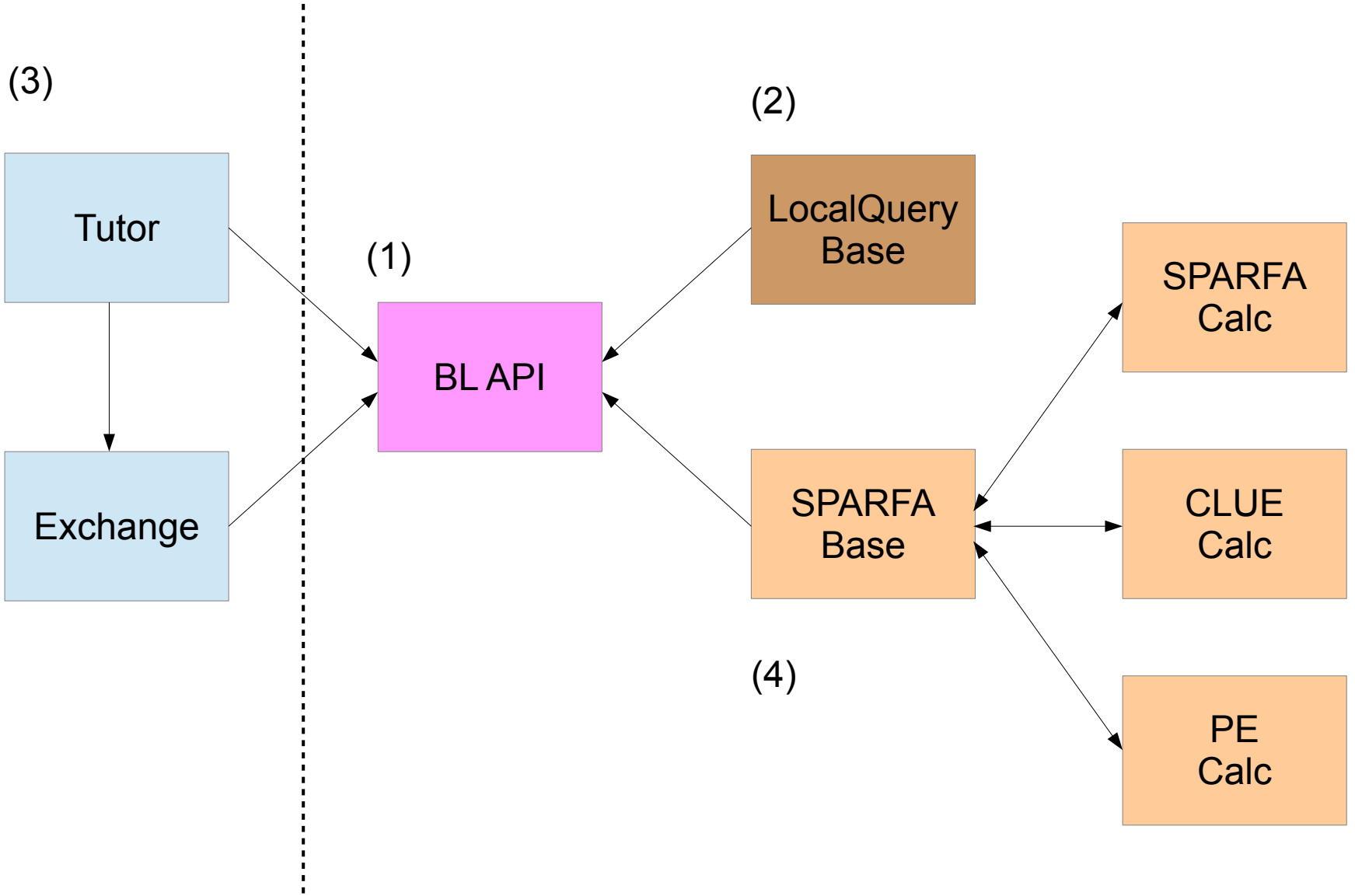

biglearn
(worker bees)
biglearn-internal
ELB
(?)
rake
ELB
(?)
Tutor
server
rake
server
Algorithm
Base 1
Exchange
server
rake
server
server
Algorithm
Base 2
server
(monitor)
BL API DB

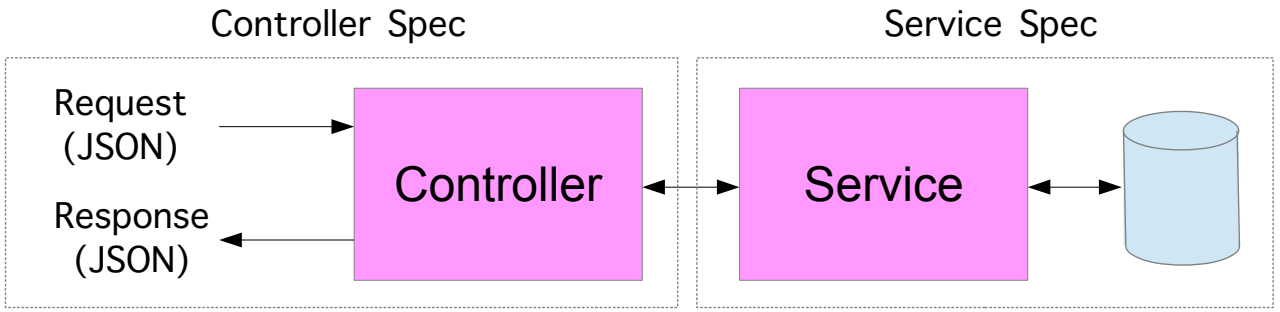
ALG BASE
(option 1)



ALG BASE
(option 2)

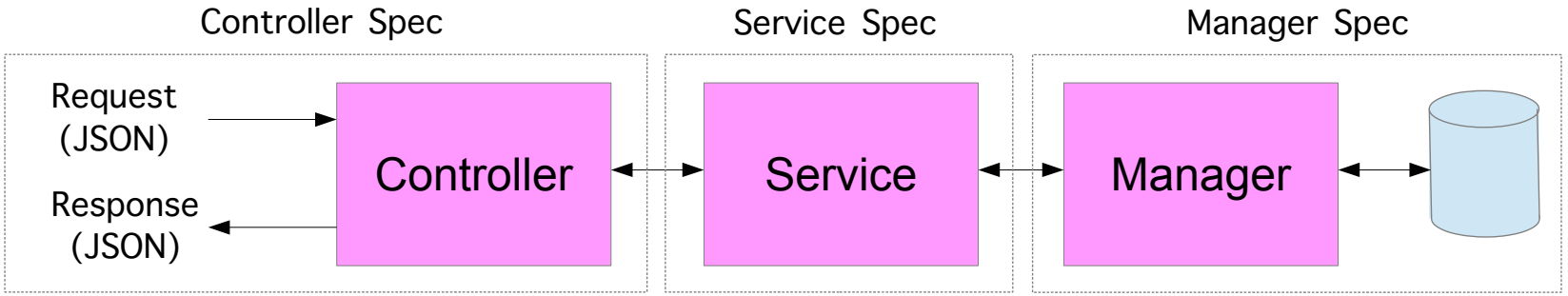



(3)
Tutor
Exchange
(1)
BL API
(2)
LocalQuery Base
SPARFA Base
SPARFA Calc
CLUE Calc
(4)
PE Calc



Request/Response JSON payload validation
passing of Request params to Service
packaging of Service results into Response

transaction creation
business logic
AR model creation and query



transaction creation
passing of params to Manager
processing of Manager results
business logic

business logic
AR model creation and query

Additionally, integration specs to cover:

- end-to-end wiring
- concurrency
- performance

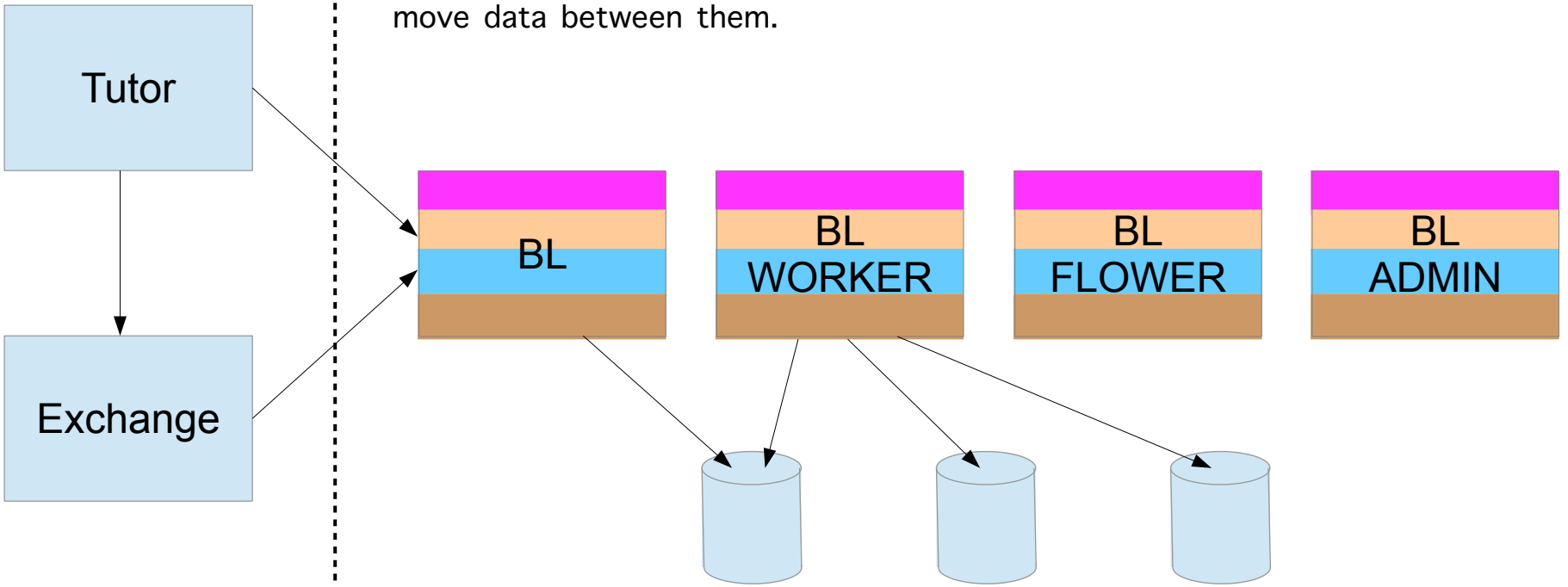
Issues that definitely need devops input:

What's the best way to describe how many machines of each type (server vs rake/python), and their associated work groups, to the deployment scripts?

How much do we want to deal with elastic scaling options, given that the request rates should be pretty well-known and constant most of the time?

How do we want to handle the situations when we know we want a temporary increase in throughput?

What's the best way to monitor the server and rake/python processes?

BIGLEARN (old)

Biglearn consists of four repos, all of which must be deployed as a unit. There are three BL databases (batch, speed, and serve) and tasks move data between them.