Enabling open-source analytics in the enterprise

Michael Lawrence (Genentech Research)

August 16, 2018

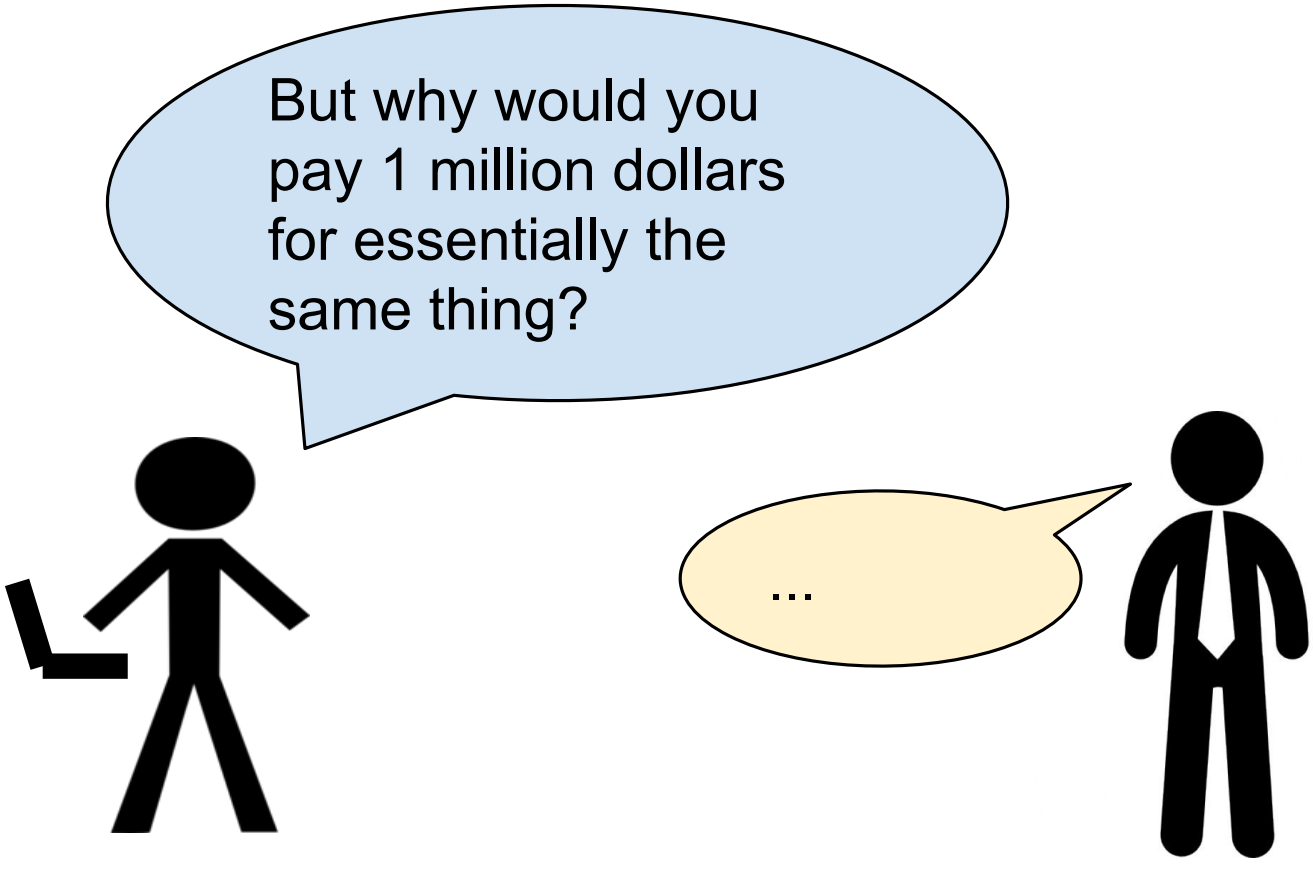
R is awesome

- R is free
- R is extensible and programmable
- R has thousands of high quality packages
  - Enable R to support virtually all research in statistical methods with minimal work by the core group
  - Enable reproducibility through versioning, ease of installation

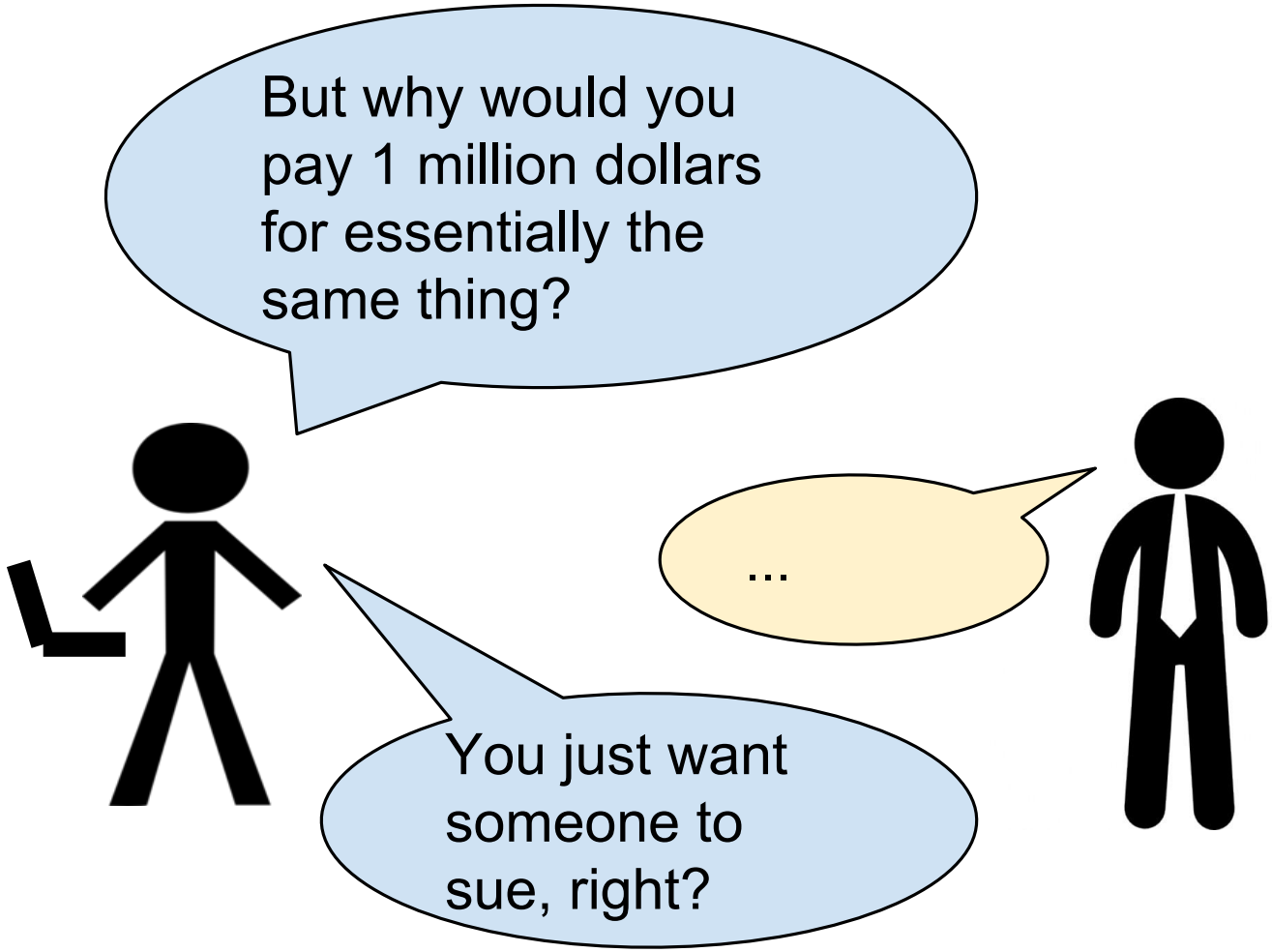
R is free!


I downloaded R onto my laptop and it was so easy to make this plot..
Yea.. but we're going with company X

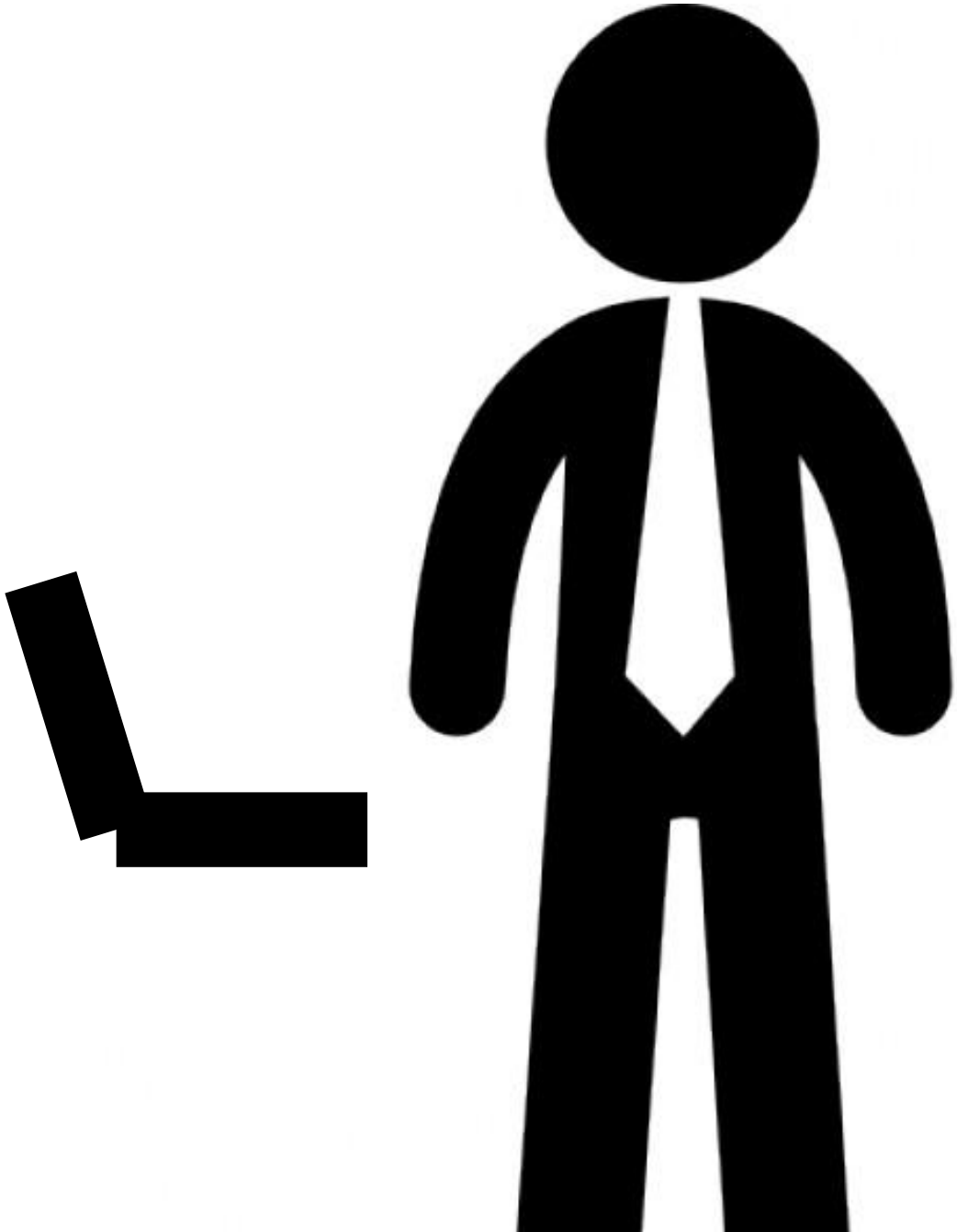
R is free!


But why would you pay 1 million dollars for essentially the same thing?
...

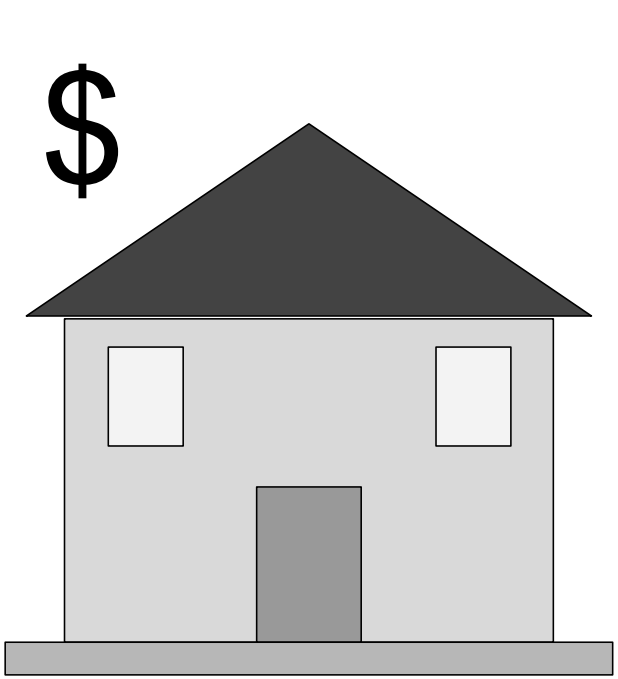
R is free!


But why would you pay 1 million dollars for essentially the same thing?
...
You just want someone to sue, right?

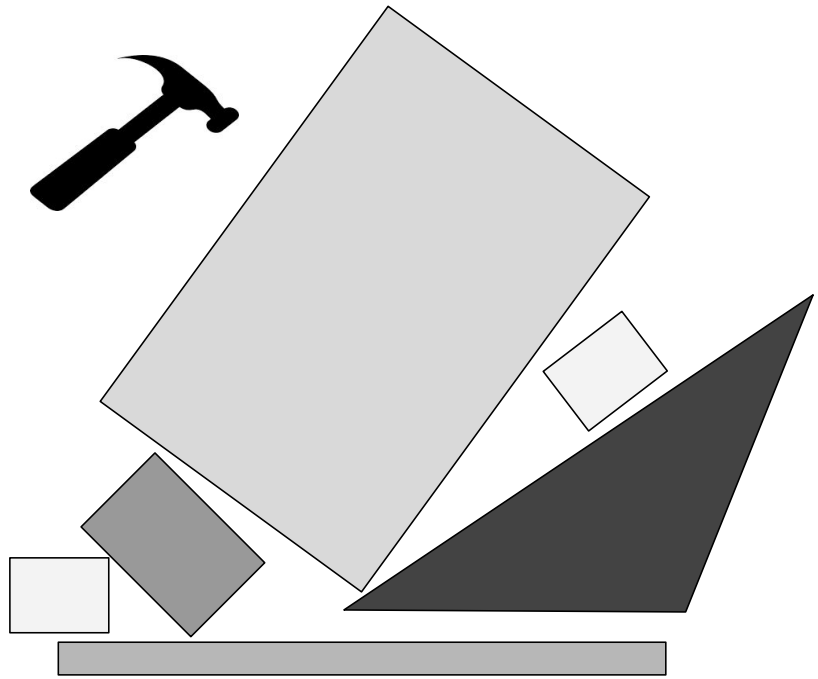
Playing devil's advocate



Build vs. buy

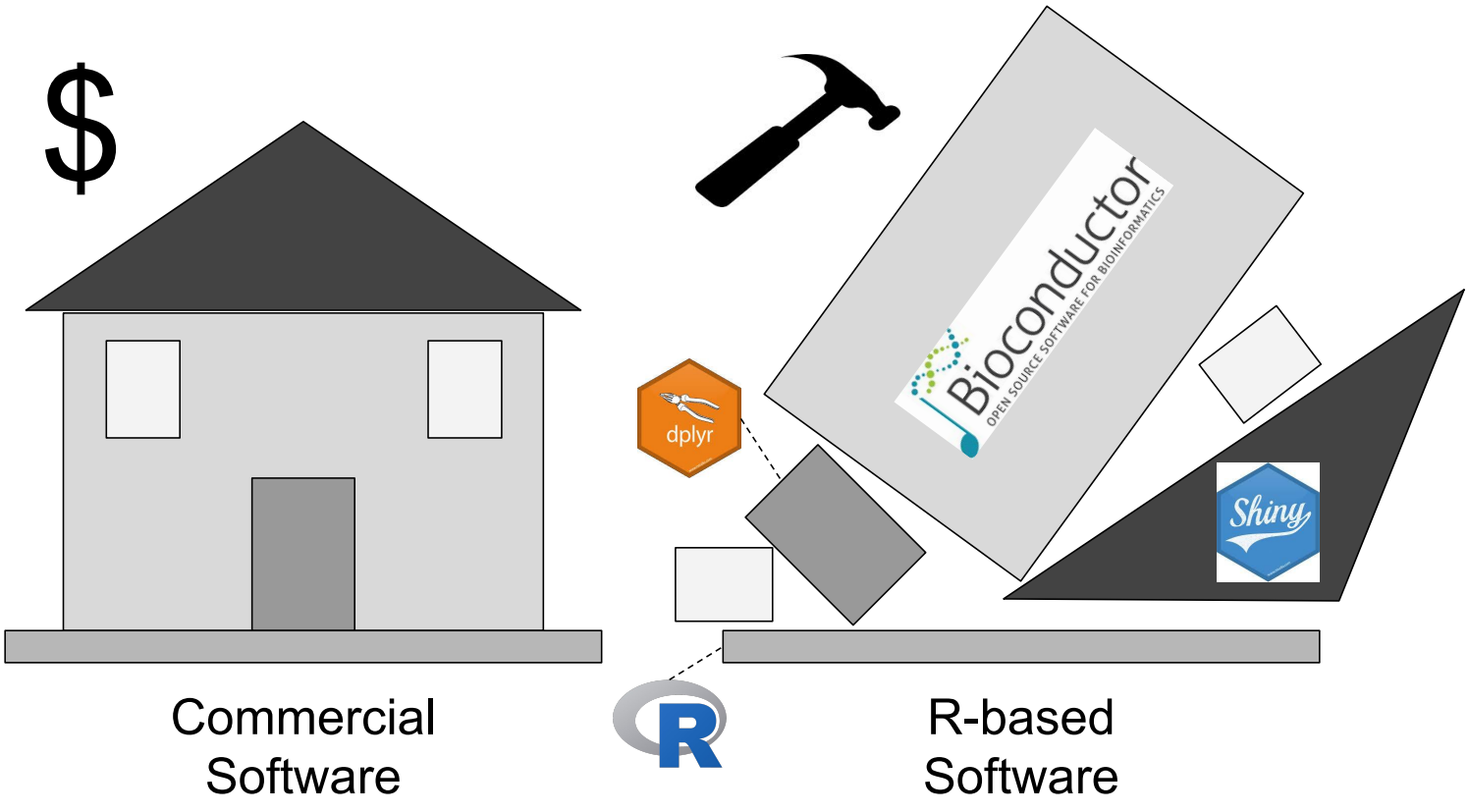


Commercial Software

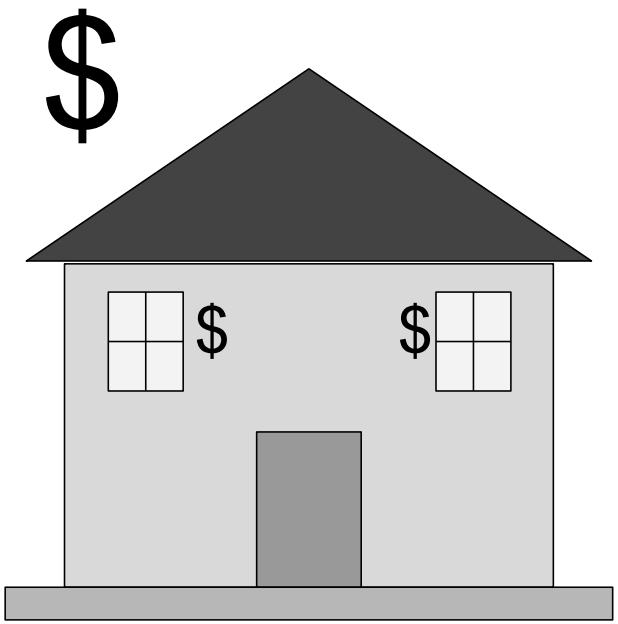


R-based Software

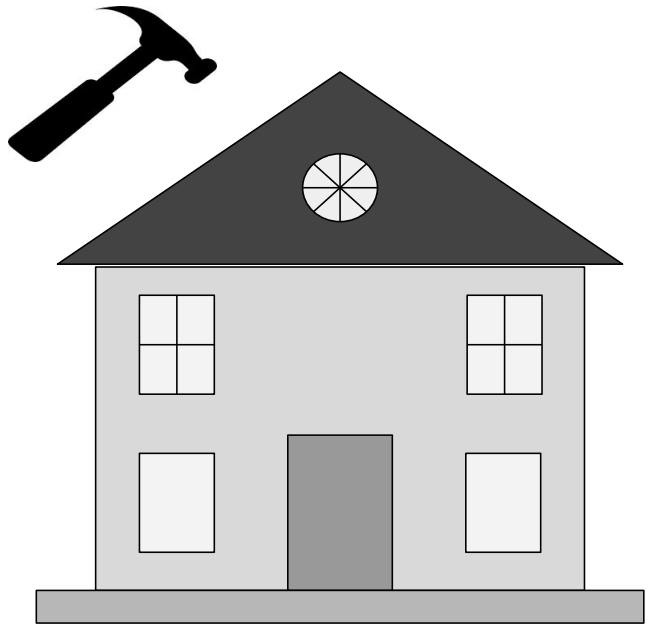
Build vs. buy


$
dplyr
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
Shiny
R
Commercial Software
R-based Software

Build vs. buy

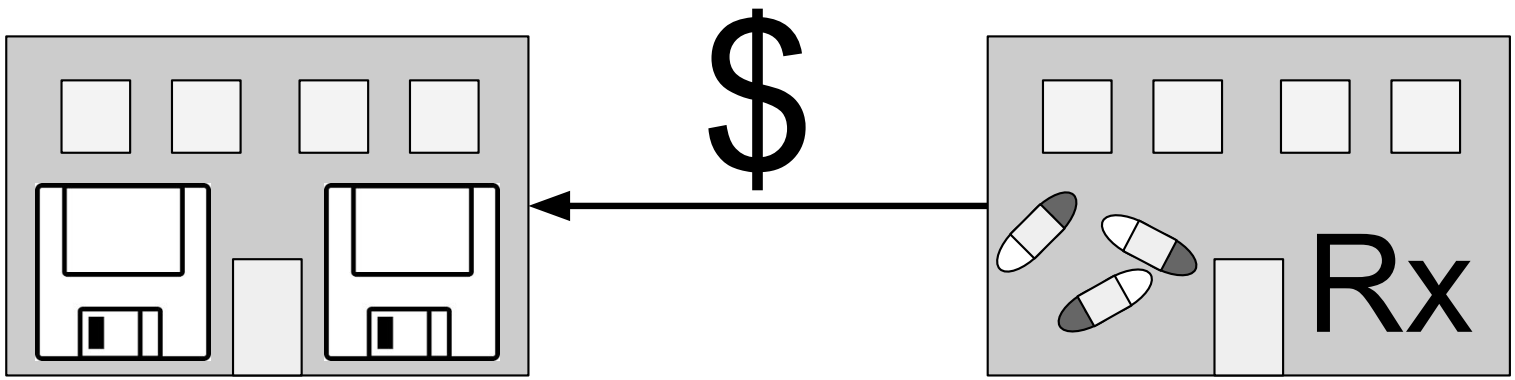


Customized
Commercial Software



Customized
R-based Software

From *paying* a software company...

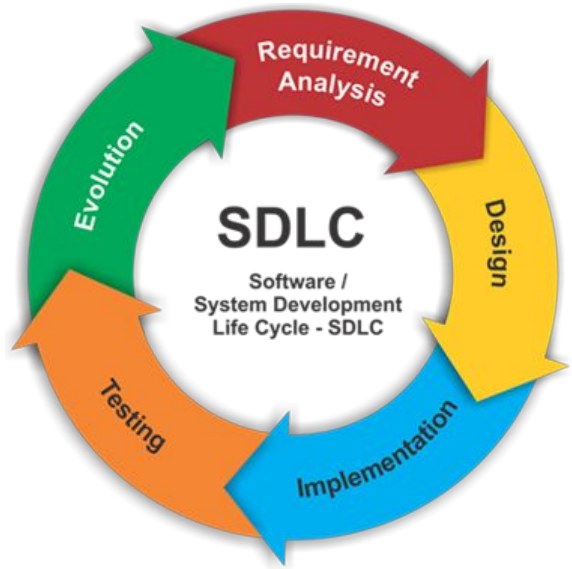


. . . to *being* a software company

R's strengths are also its weaknesses

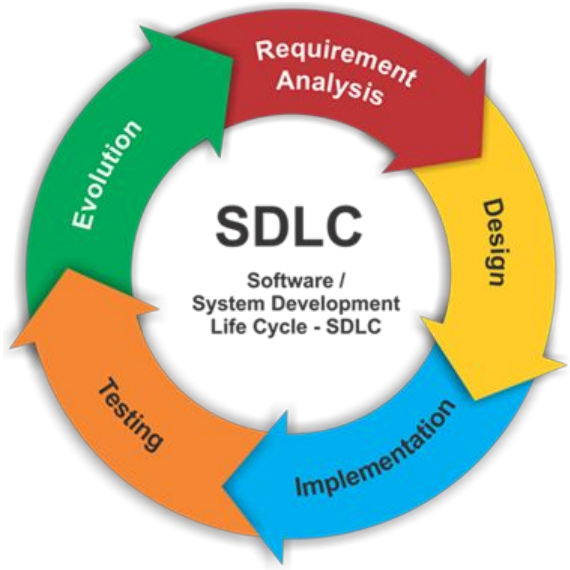
- ~~R is free~~ R is an investment
- ~~R is extensible and programmable~~ R needs to be programmed
  - How do we develop software for science?
    - That enables collaboration across the scientific enterprise?
- ~~R has thousands of high quality packages~~ R has too many packages
  - How do we manage our R environment?

Gated evolution: The scientific software lifecycle



Gated evolution: The scientific software lifecycle

Inception
(during scientific inquiry)

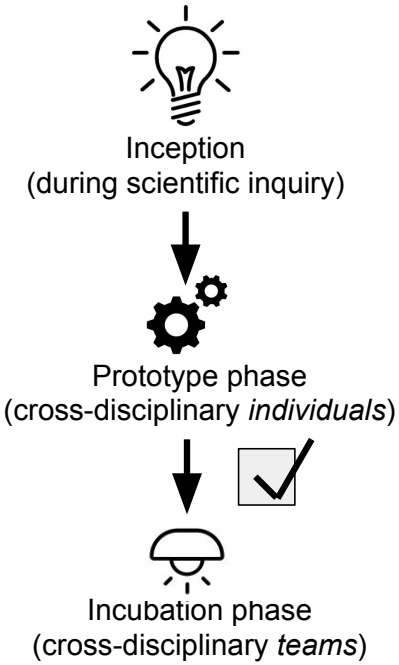


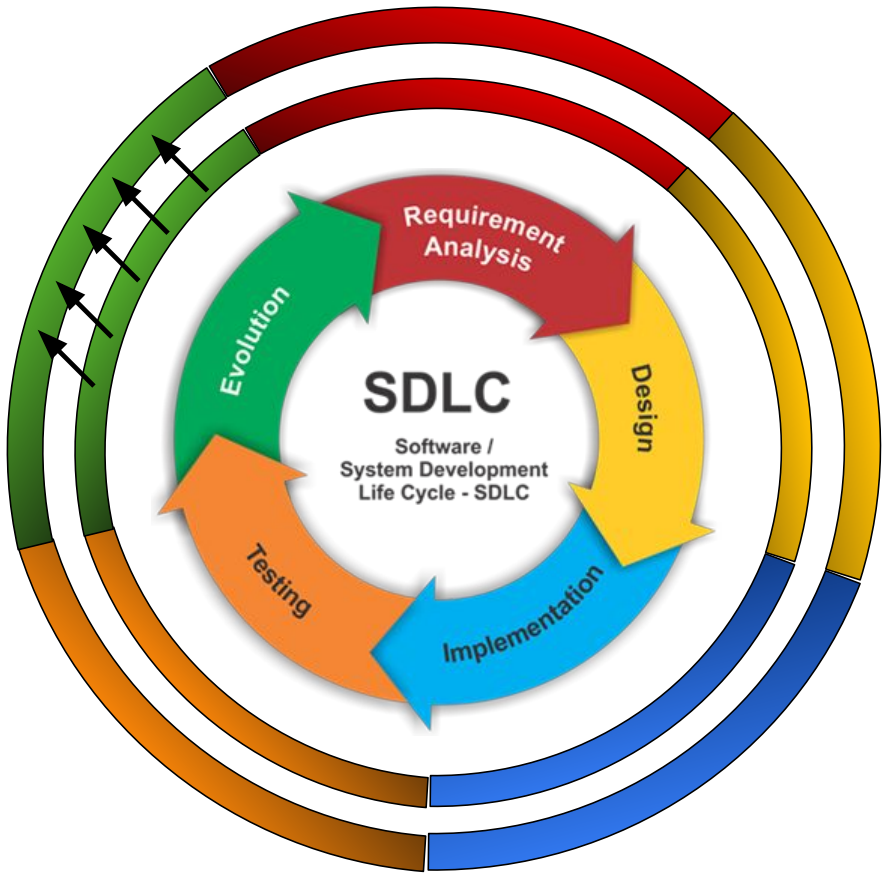
Gated evolution: The scientific software lifecycle

Inception
(during scientific inquiry)

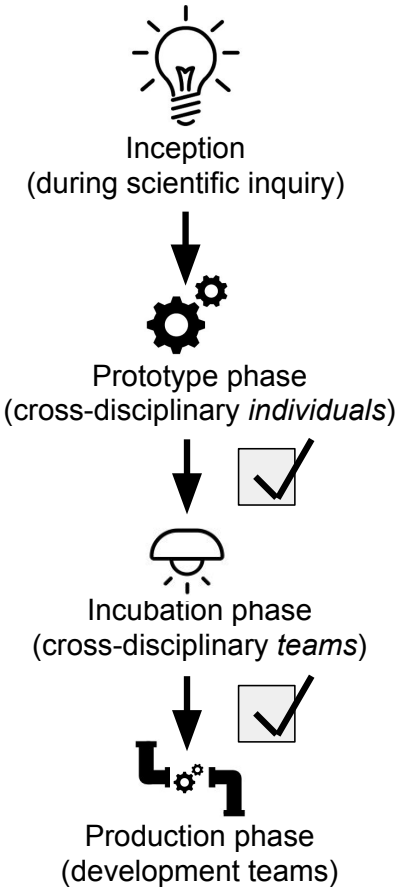
Prototype phase
(cross-disciplinary *individuals*)

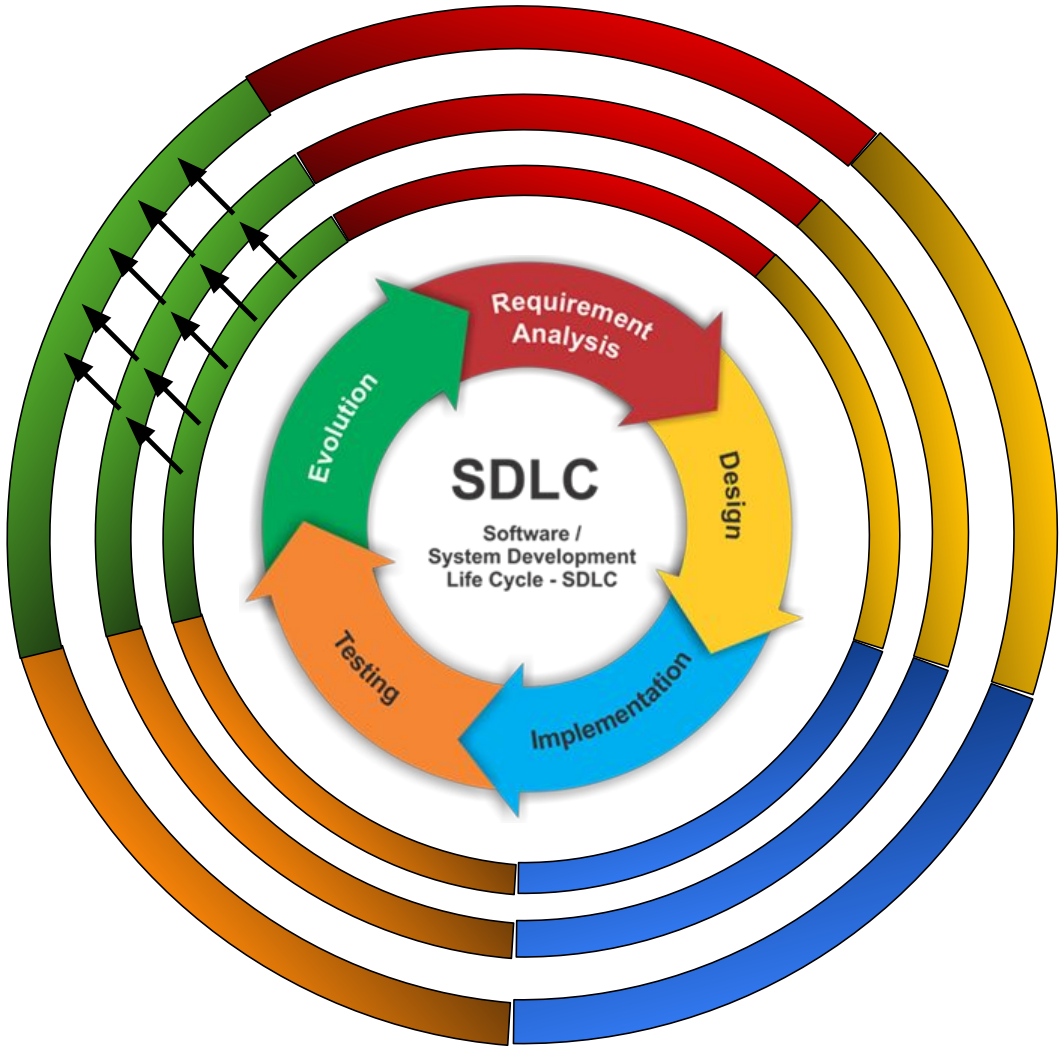
Gated evolution: The scientific software lifecycle



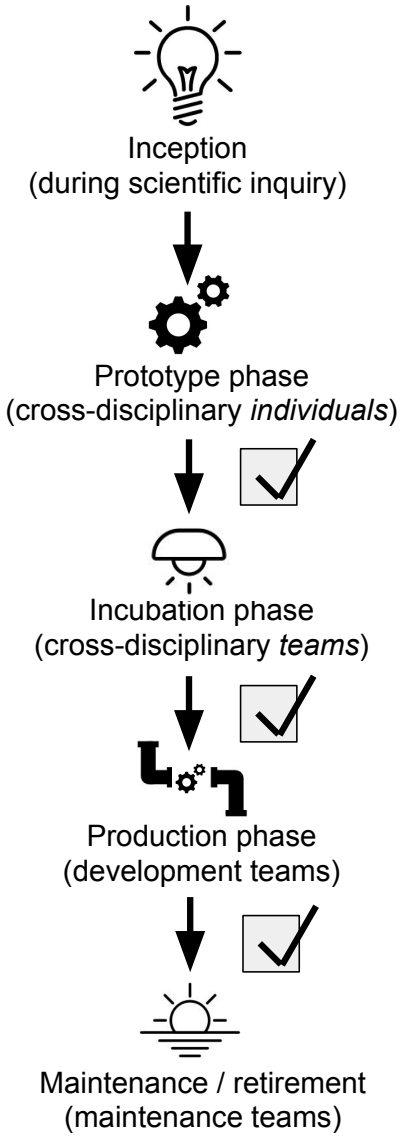


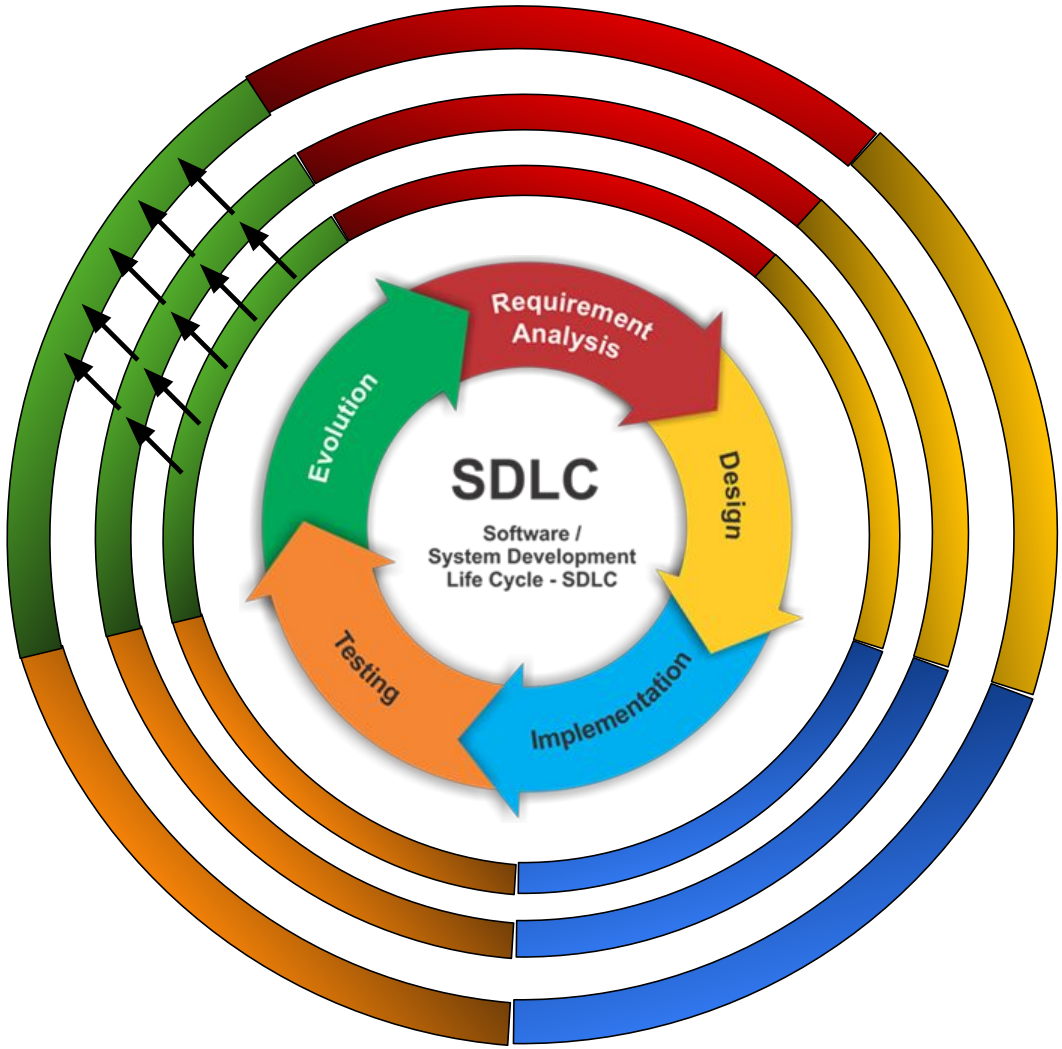
Gated evolution: The scientific software lifecycle



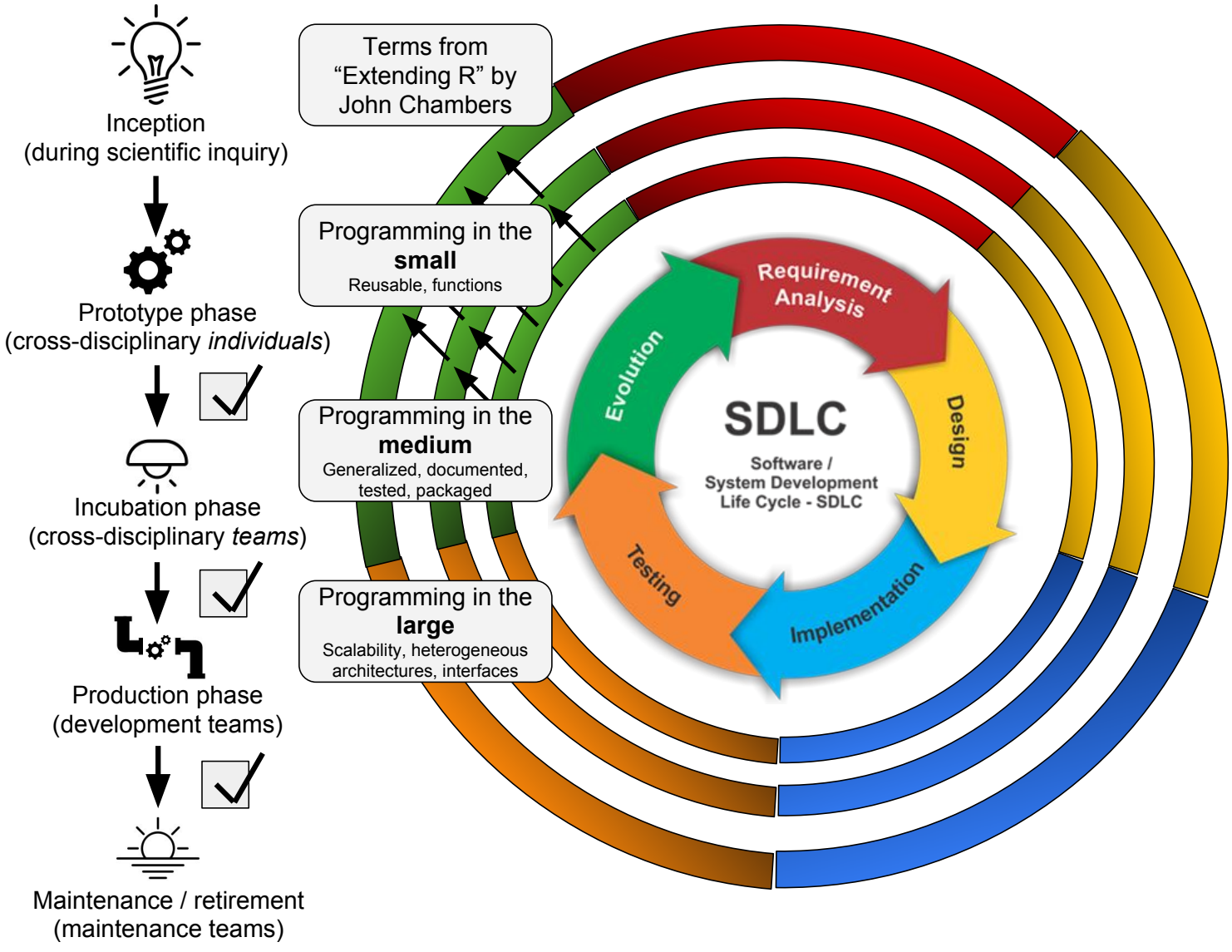


Gated evolution: The scientific software lifecycle

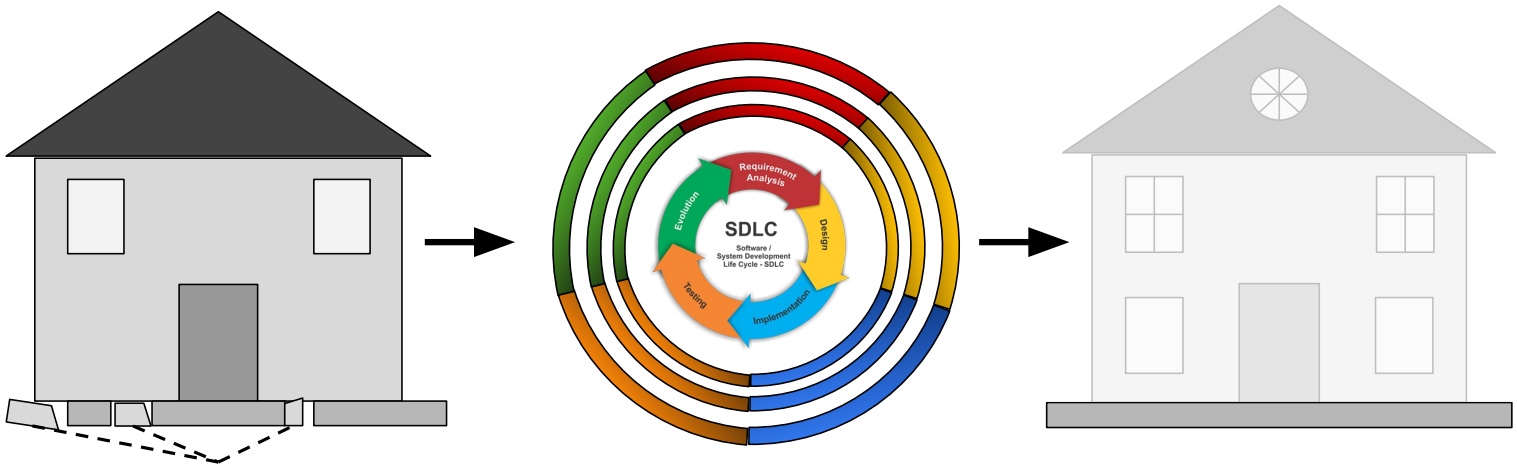




Gated evolution: The scientific software lifecycle



The incubator feedback loop

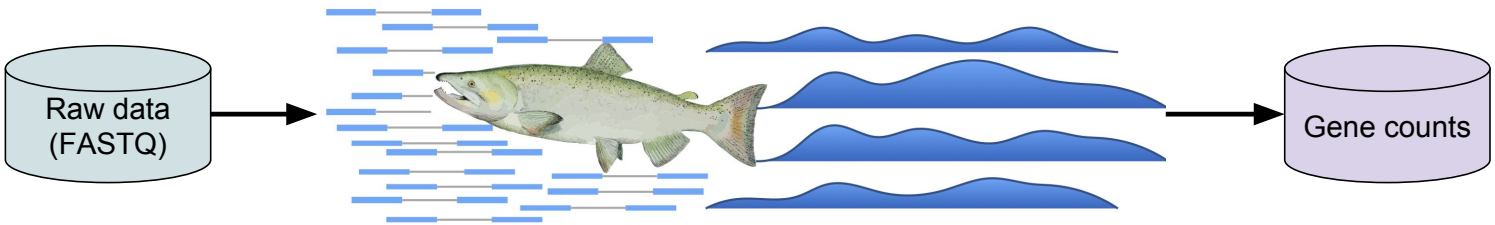


Infrastructure gaps

Incubation Process

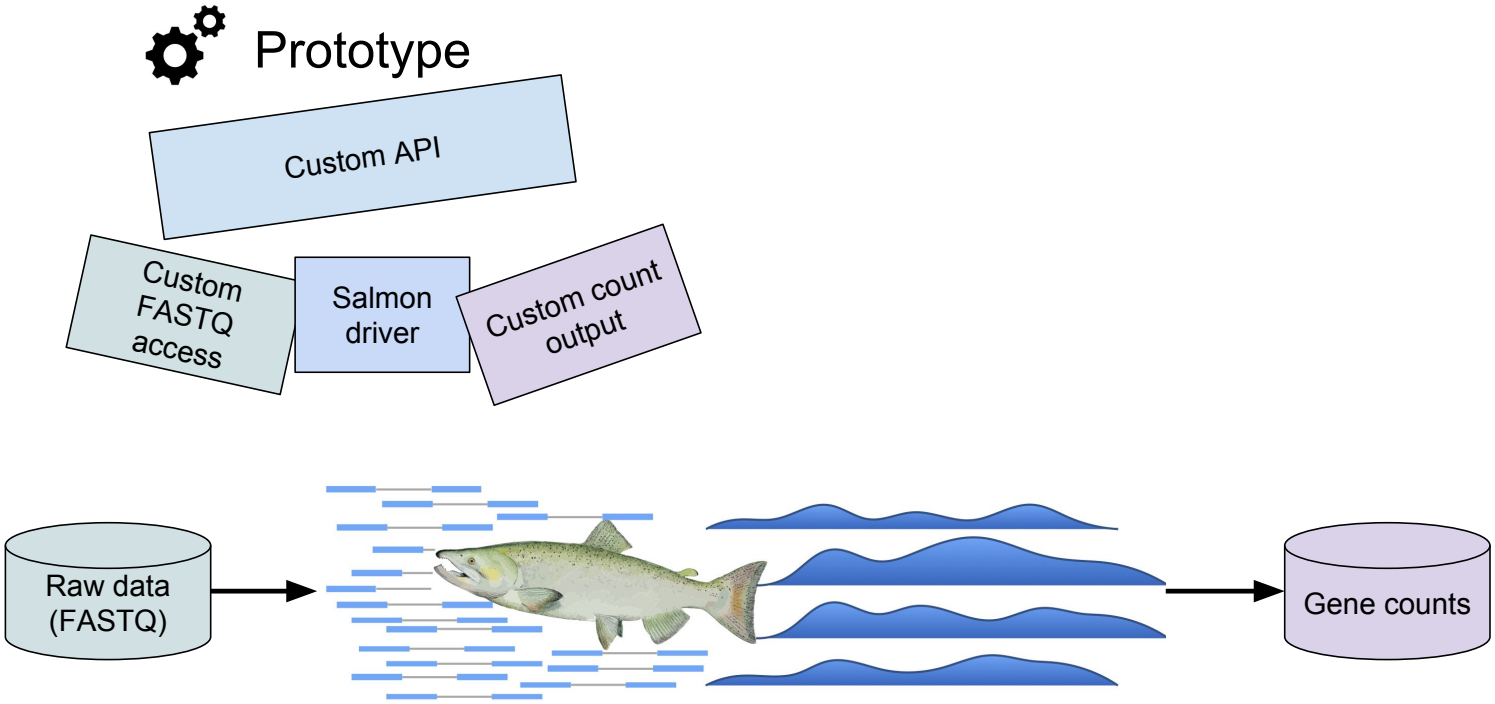
Gaps filled for future prototypes

Incubation example: gSalmon



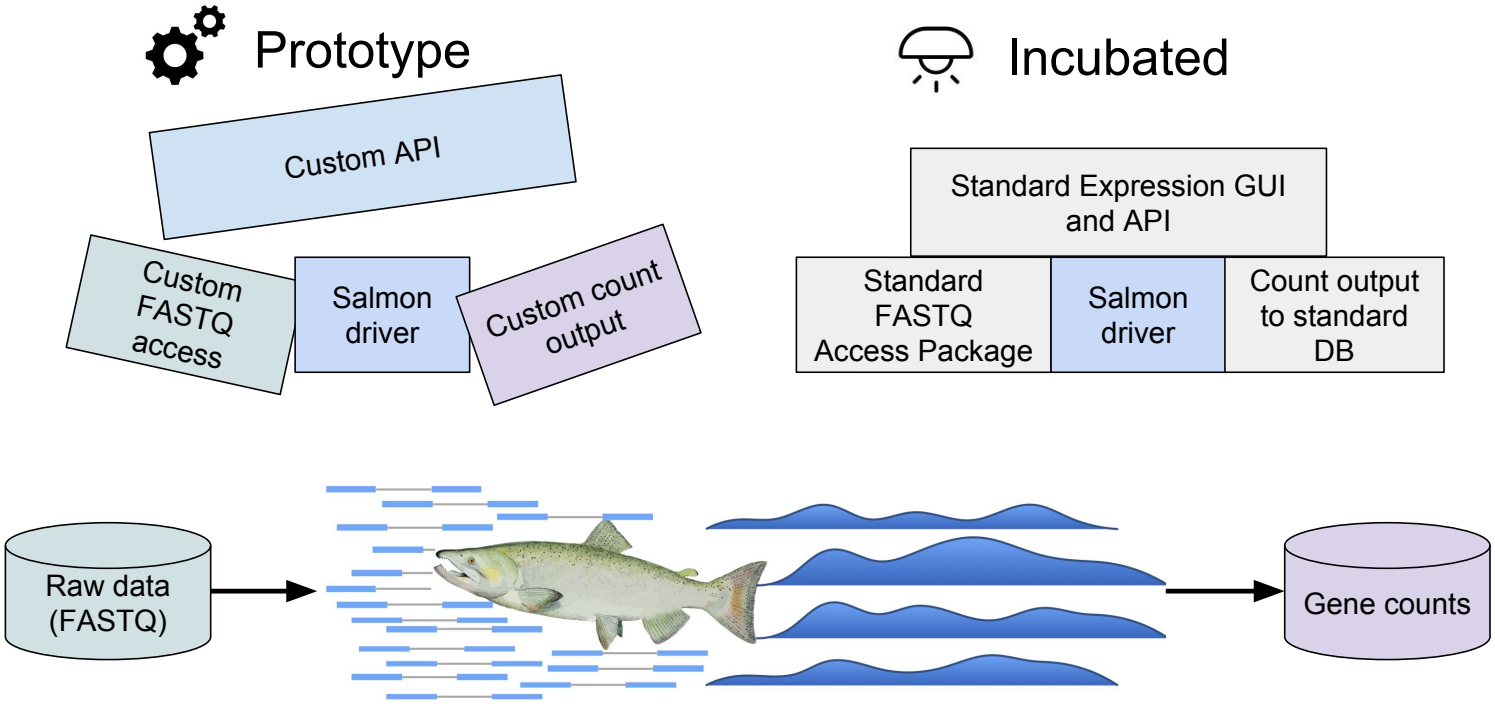
Salmon (Public tool for RNA-seq expression counting)

Incubation example: gSalmon



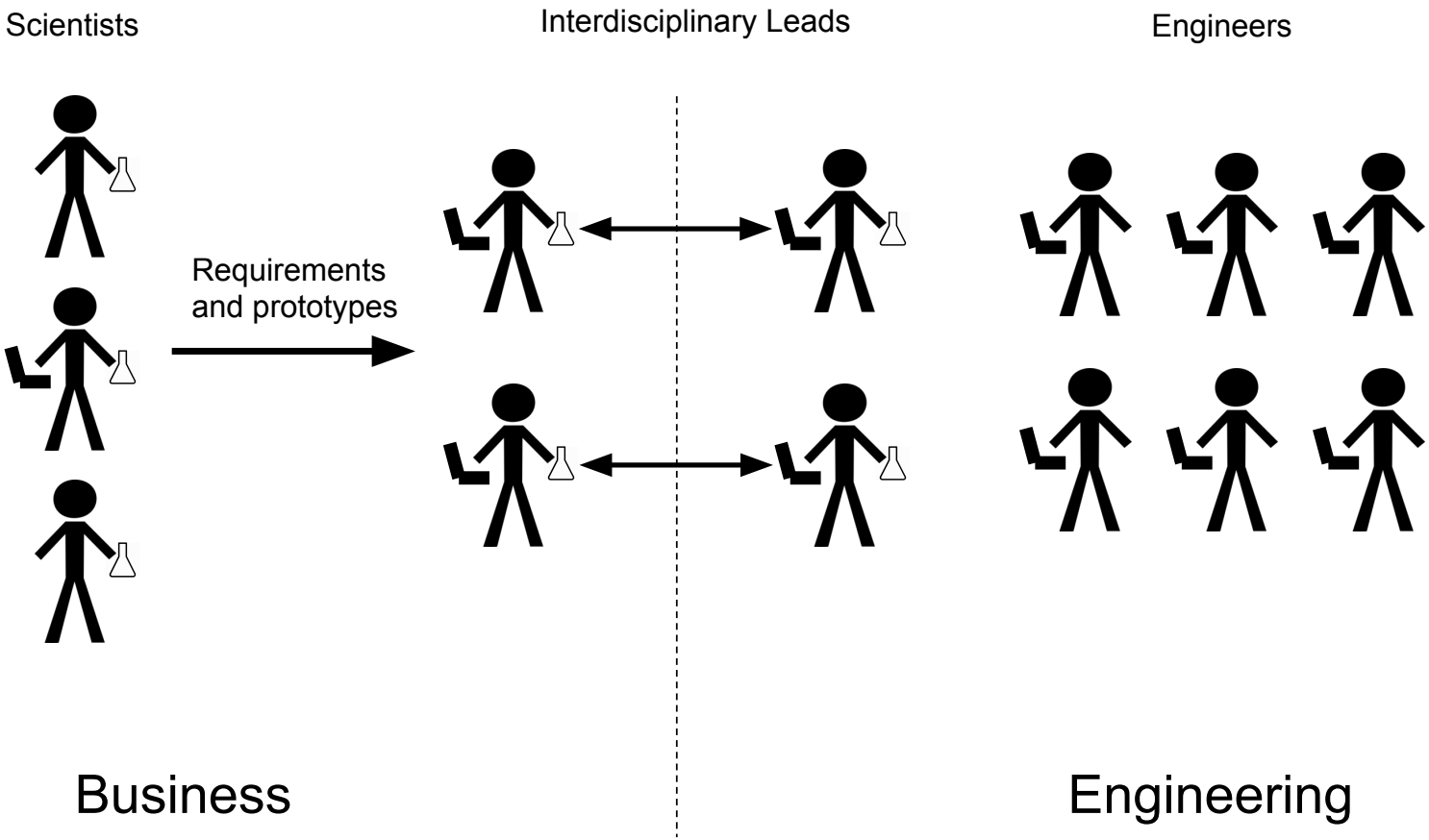
Salmon (Public tool for RNA-seq expression counting)

Incubation example: gSalmon

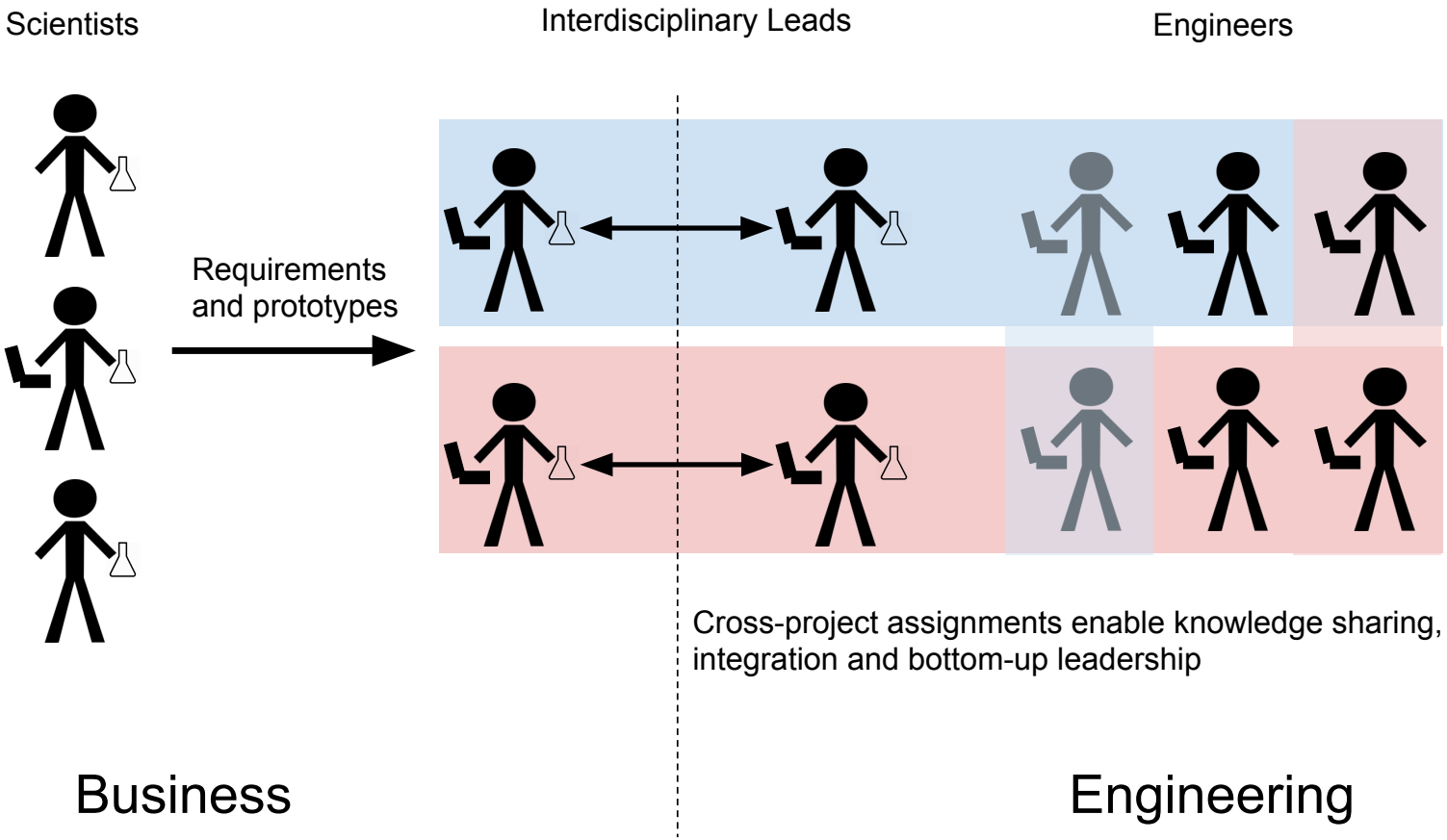


Salmon (Public tool for RNA-seq expression counting)

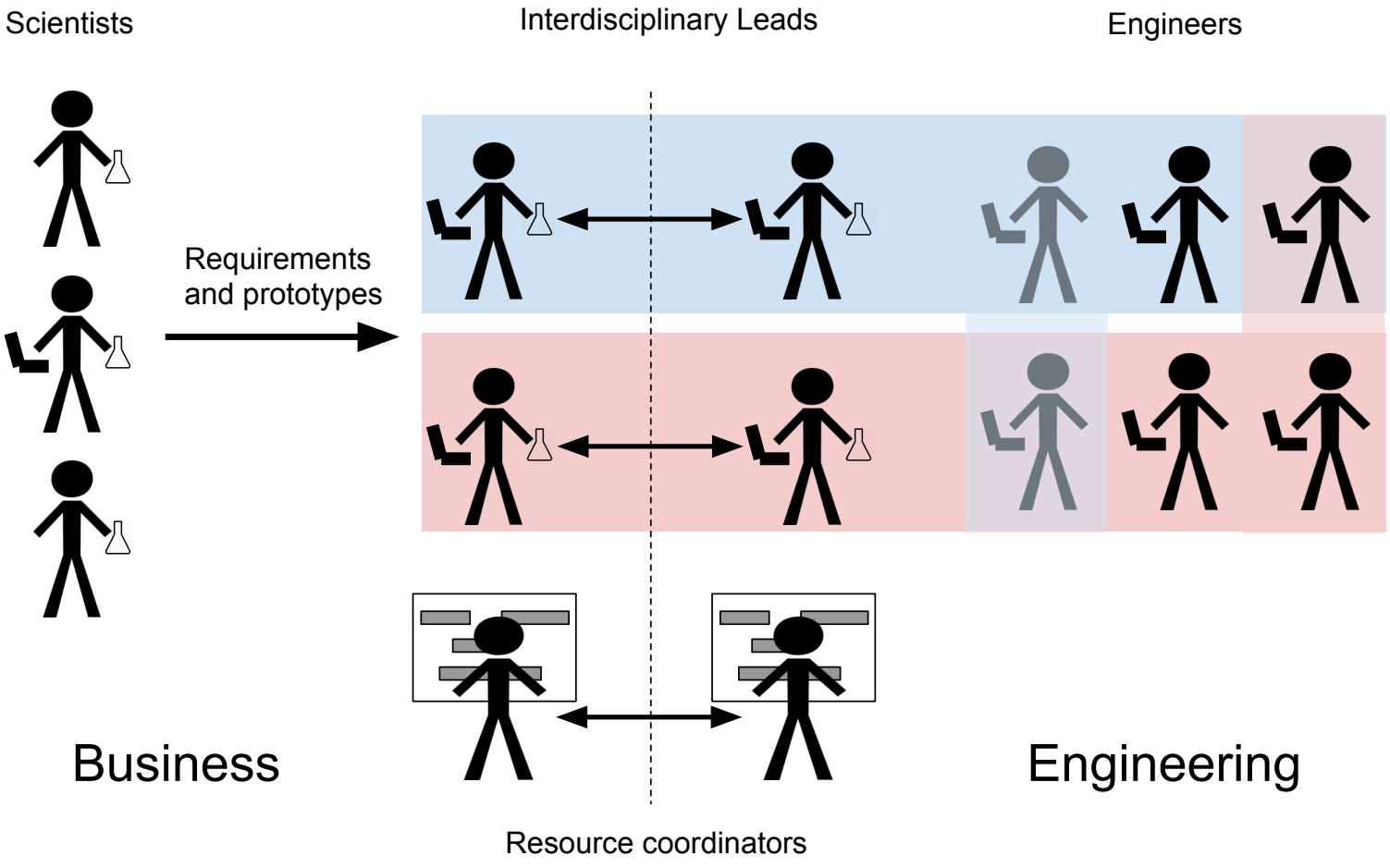
Managing developer resources



Managing developer resources



Managing developer resources



Contributing back to open-source

the engineering case

- Setting out to open-source puts the right perspective on development and leads to better software
- Lets us steer open-source development:
  - Specific projects
  - Trend setting, across the field

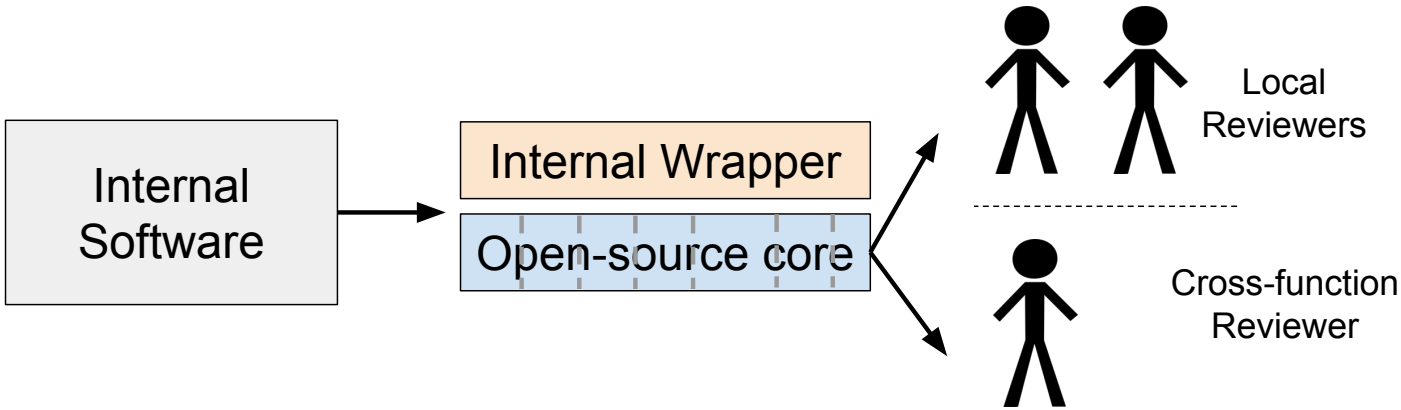
Contributing back to open-source

the scientific case

- Open-source software publication is an intellectual contribution to science, equivalent to a paper
- Enhances potential for collaborations
- Enables reproducibility of published results

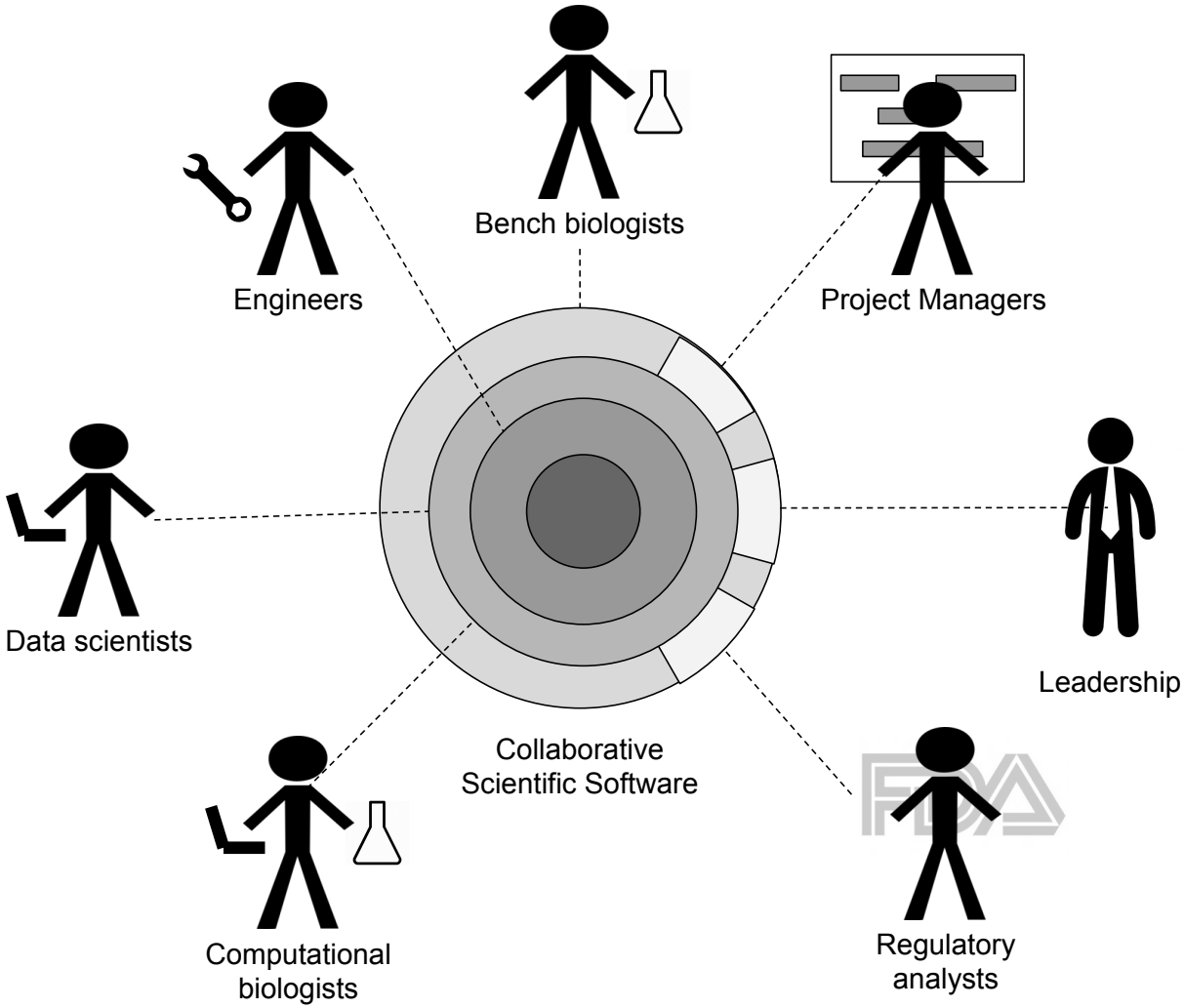
Open-sourcing policy and process

- Decide to open-source early in incubation process
  - Potentially separate open-source and in-house codebases
  - Most infrastructure will need to be open-source, in order to support open-sourcing of dependents
- Piggyback on existing manuscript publication process
  - Code review in place of reviewing text
  - Encourage cross-functional reviews for learning and potential reuse/unification of software
- Develop understanding and standard practices around licensing
- Adopt open-source culture internally (*inner sourcing*)



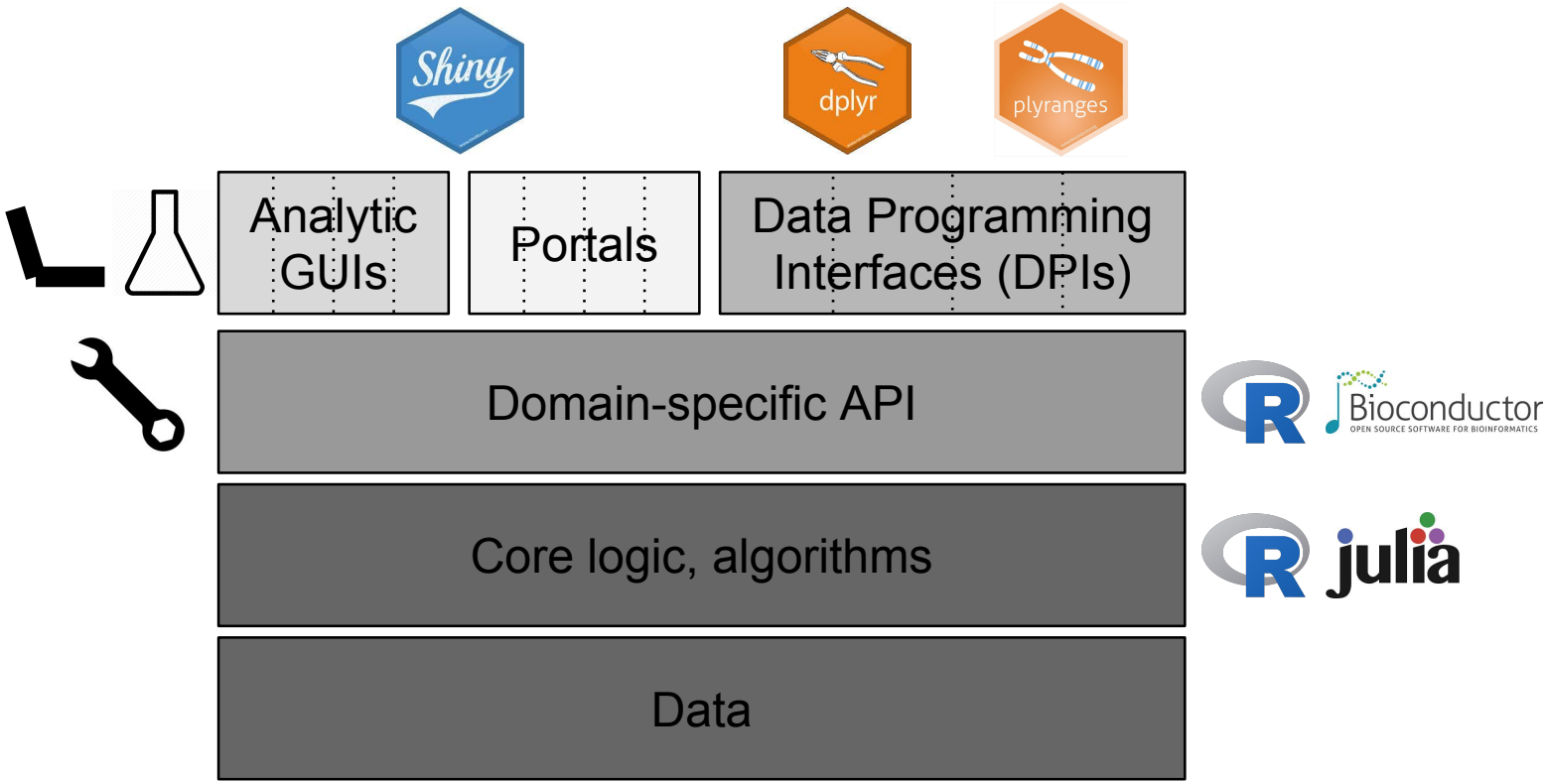
Science is collaborative

so our software should integrate multiple perspectives



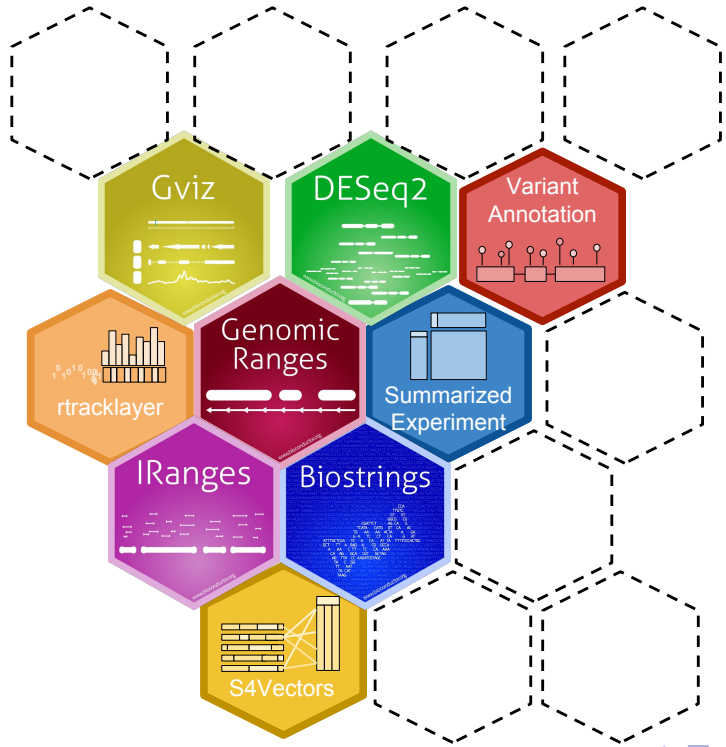
The general scientific software stack

An interface for every role

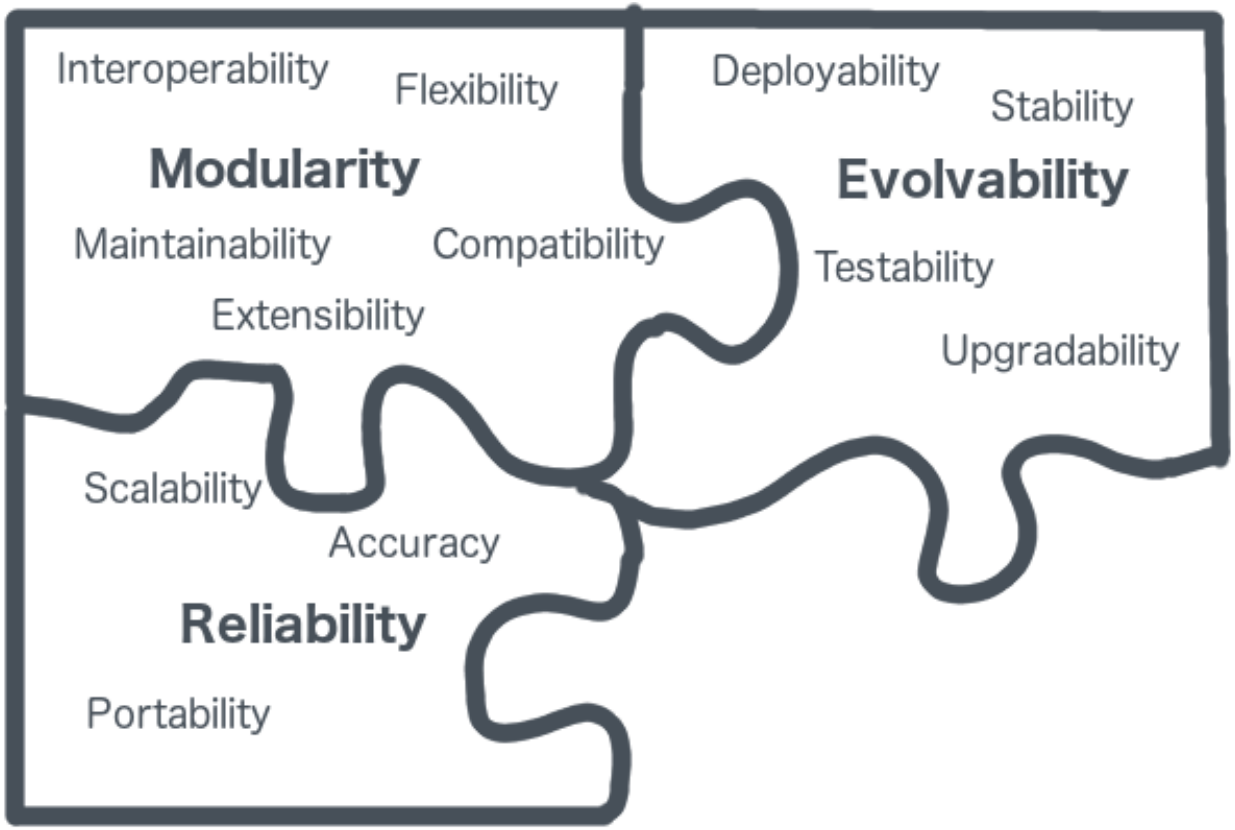


Bioconductor

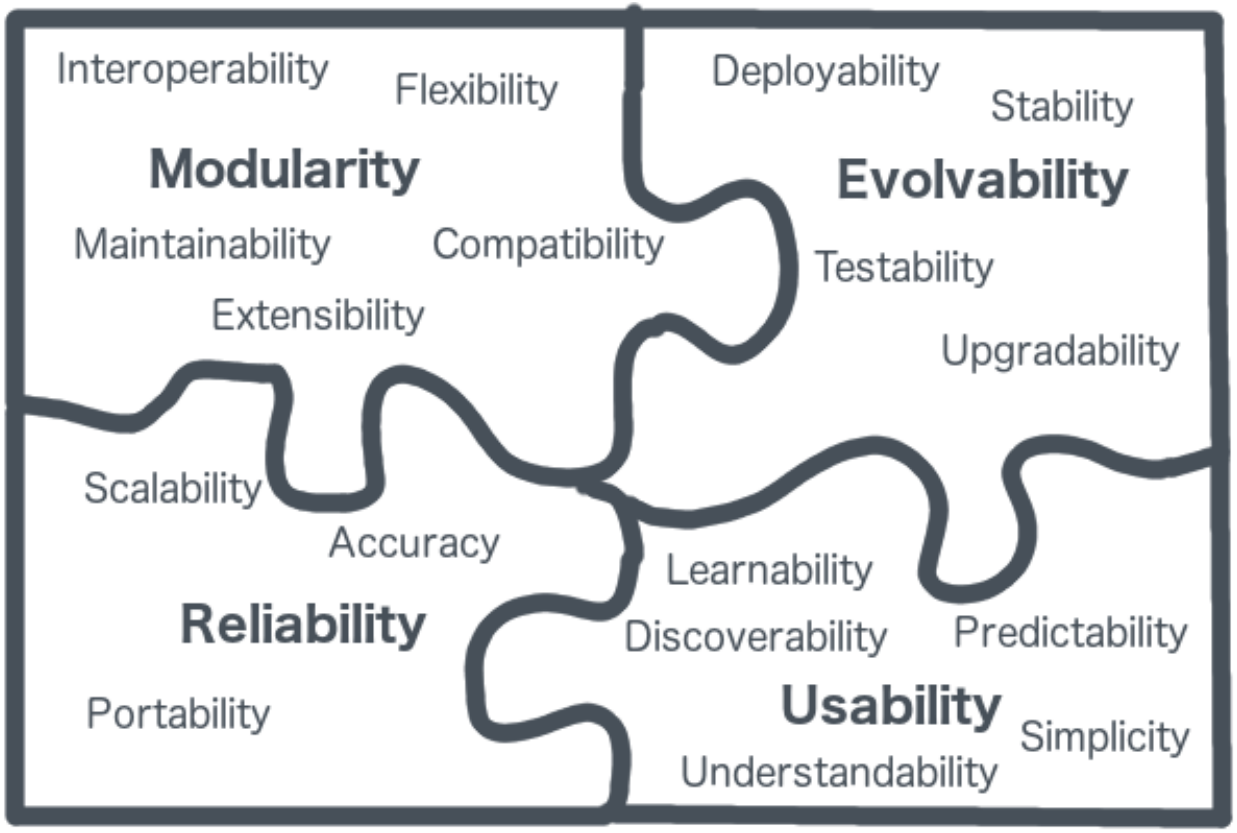
- Extends R with an integrated platform for exploratory genomics
- Implements standardized, scalable, interoperable data structures
- Hosts state-of-the-art algorithms
- >1500 community-contributed packages



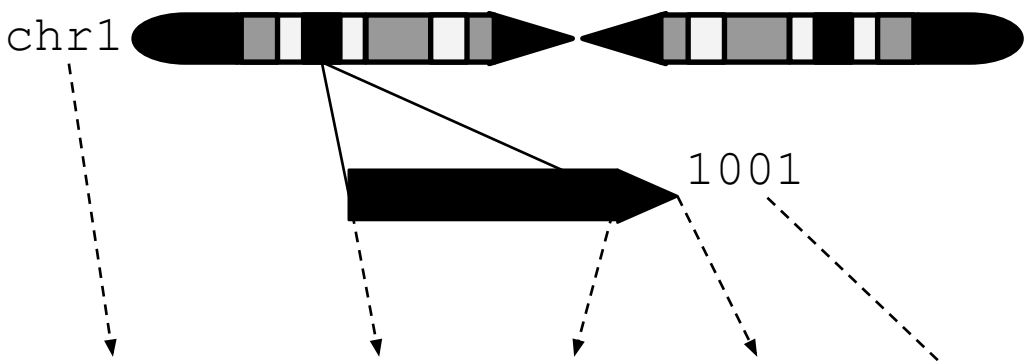
Usability: the missing piece



Usability: the missing piece



Genomic Ranges



seqname	start	end	strand	gene_id	score
chr1	102012	120303	+	1001	10
chr1	520211	526211	-	2151	25
...	...	...	...	...	...

plyranges: a grammar for genomic data manipulation

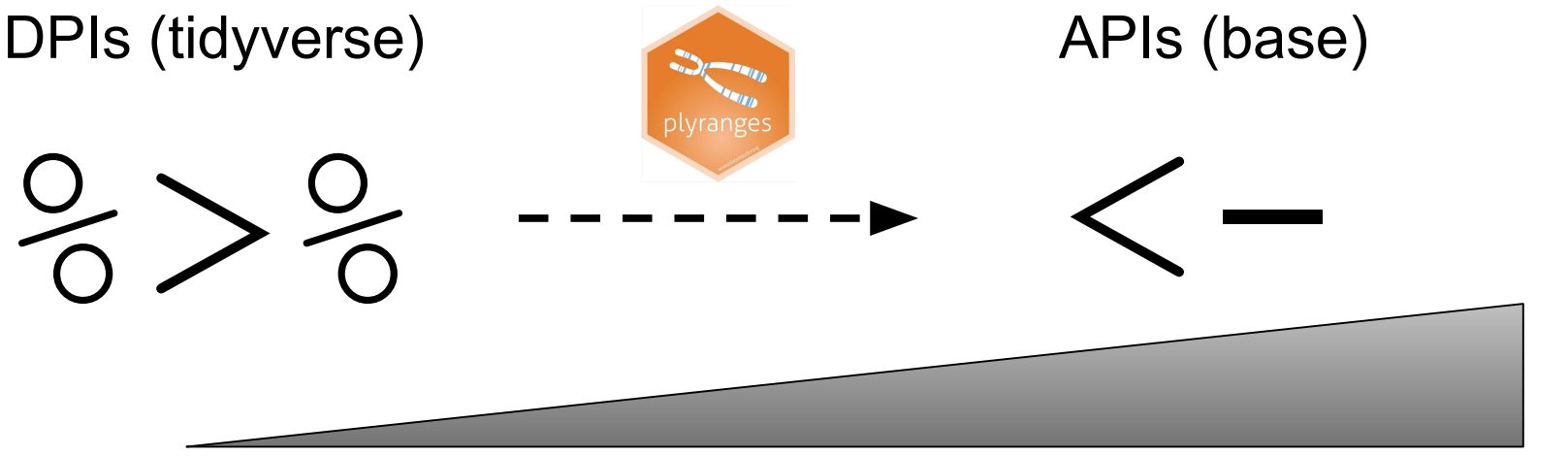
- Extension of relational algebra and dplyr syntax to genomics
- Operations are:
  - Cohesive (do a single thing)
  - Endomorphic (`<T> <? extends T> f(T x)`)
  - Verb-oriented in syntax
- Based core Bioconductor data structures:

  GRanges represents annotated genomic ranges

  SummarizedExperiment coordinates experimental assay data with sample and feature annotations
- Fluency emerges from chaining of verbs

```
compute_coverage("a.bam") %>%
    filter(score > 0) %>%
    intersect(read_bed("b.bed"))
```

Formal data structures enable transition to programming



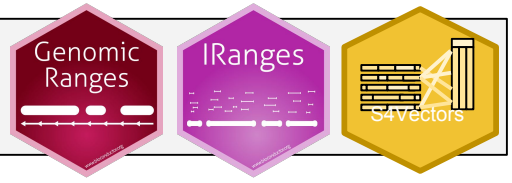
Programming in the **small**
Reusable, functions

Programming in the **medium**
Generalized, documented, tested, OOP, packaged

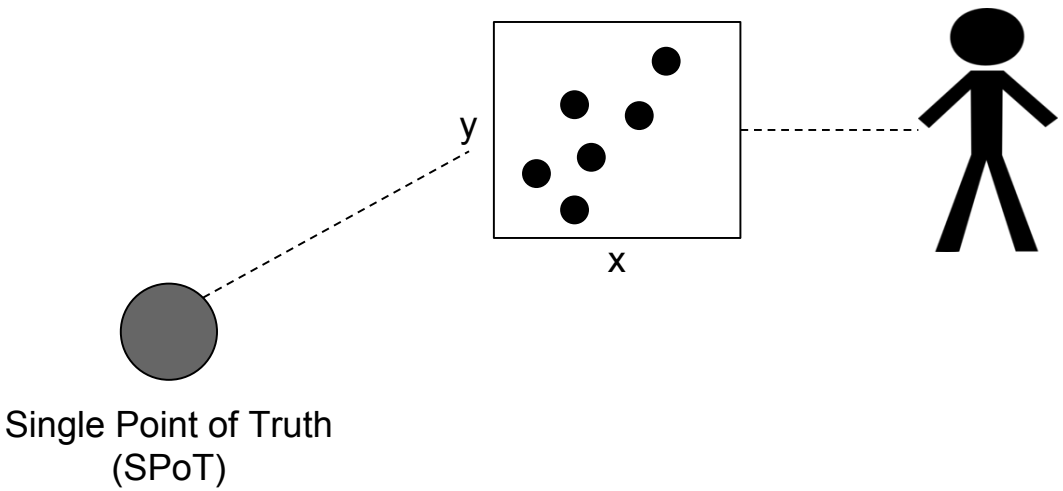
Programming in the **large**
Scalability, heterogeneous architectures, interfaces

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

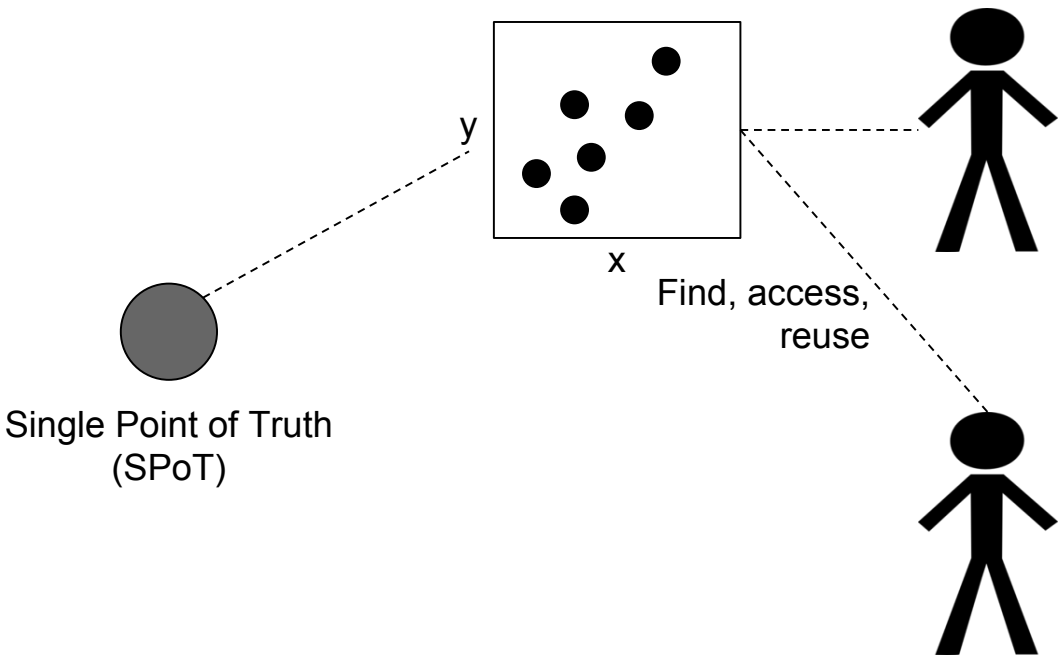
Interoperable data structures



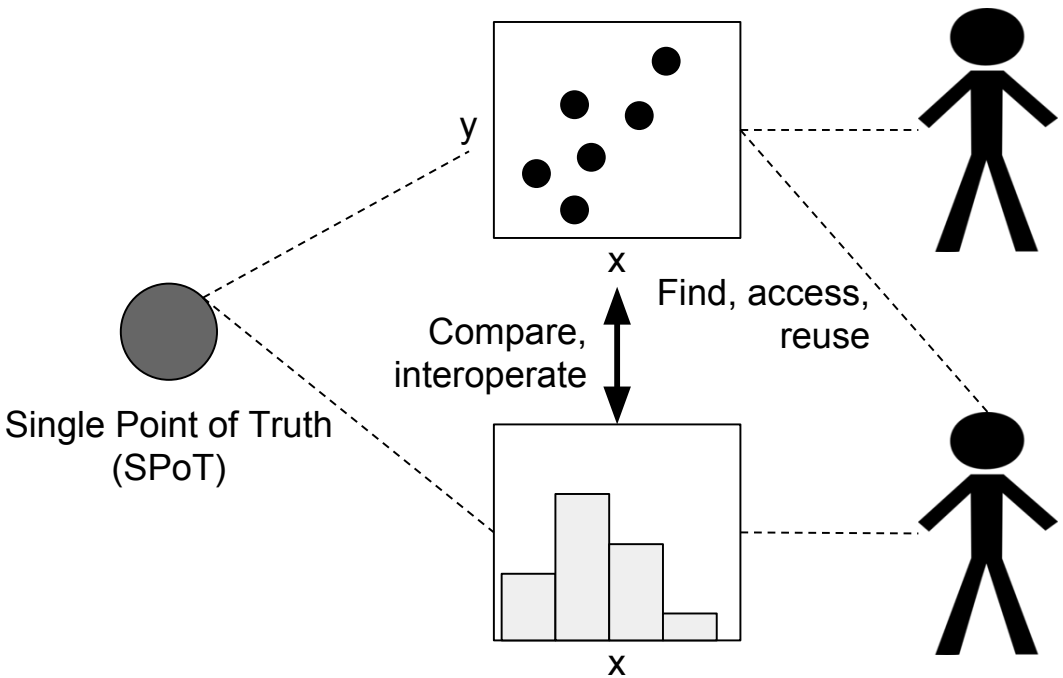
Single points of truth enhance collaboration



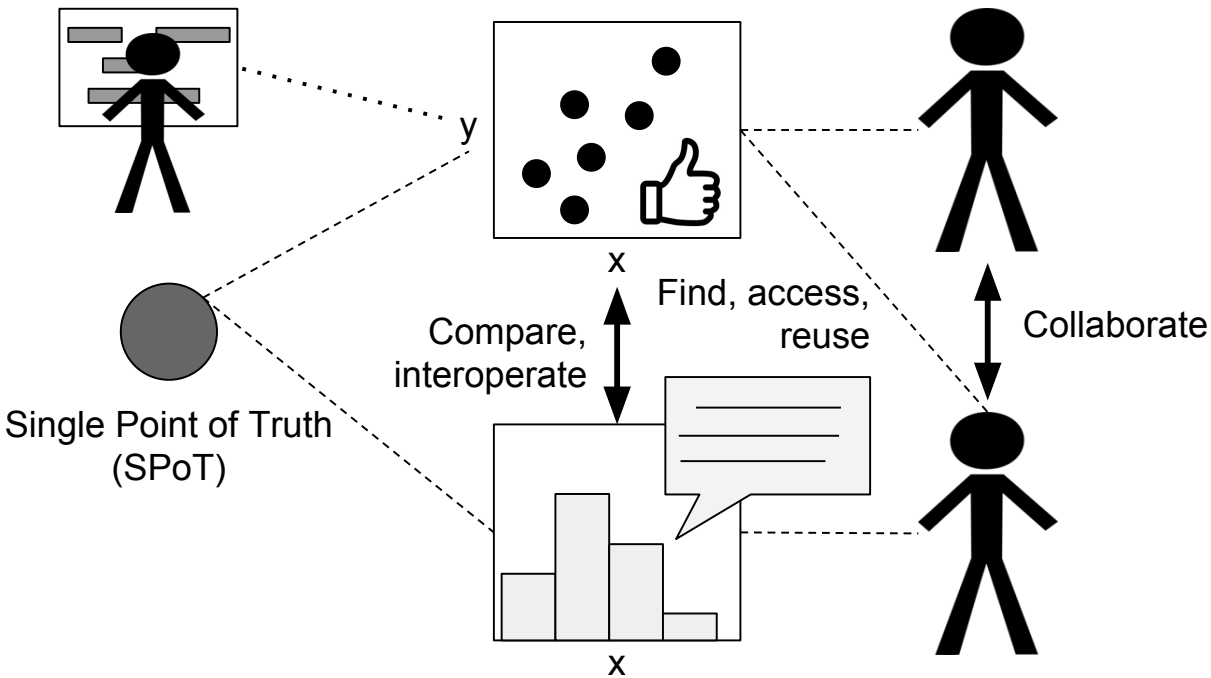
Single points of truth enhance collaboration



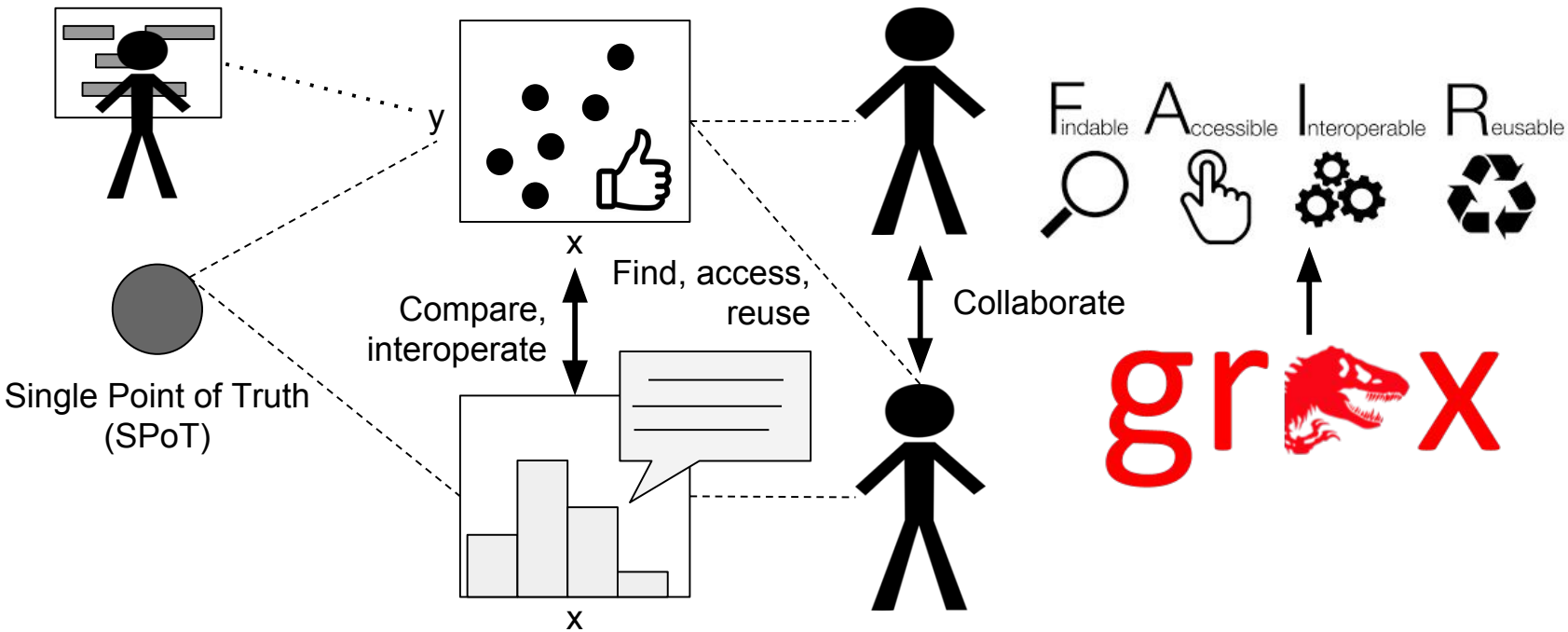
Single points of truth enhance collaboration



Single points of truth enhance collaboration



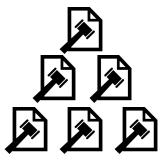
Single points of truth enhance collaboration



Requirements of a data analysis environment

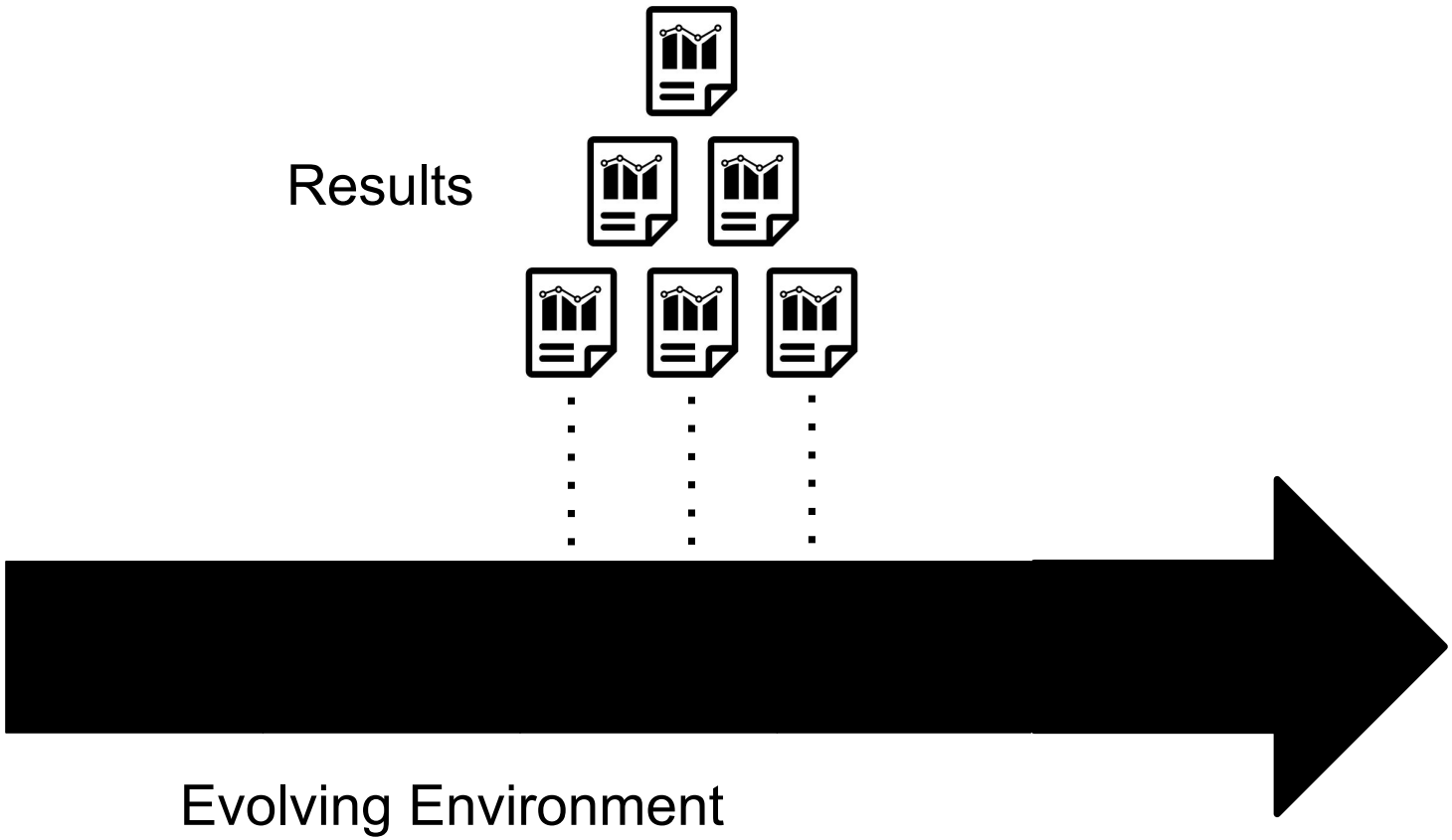
Metered evolution,

- Cohort-based build, test, deploy
- Discover tools and applications

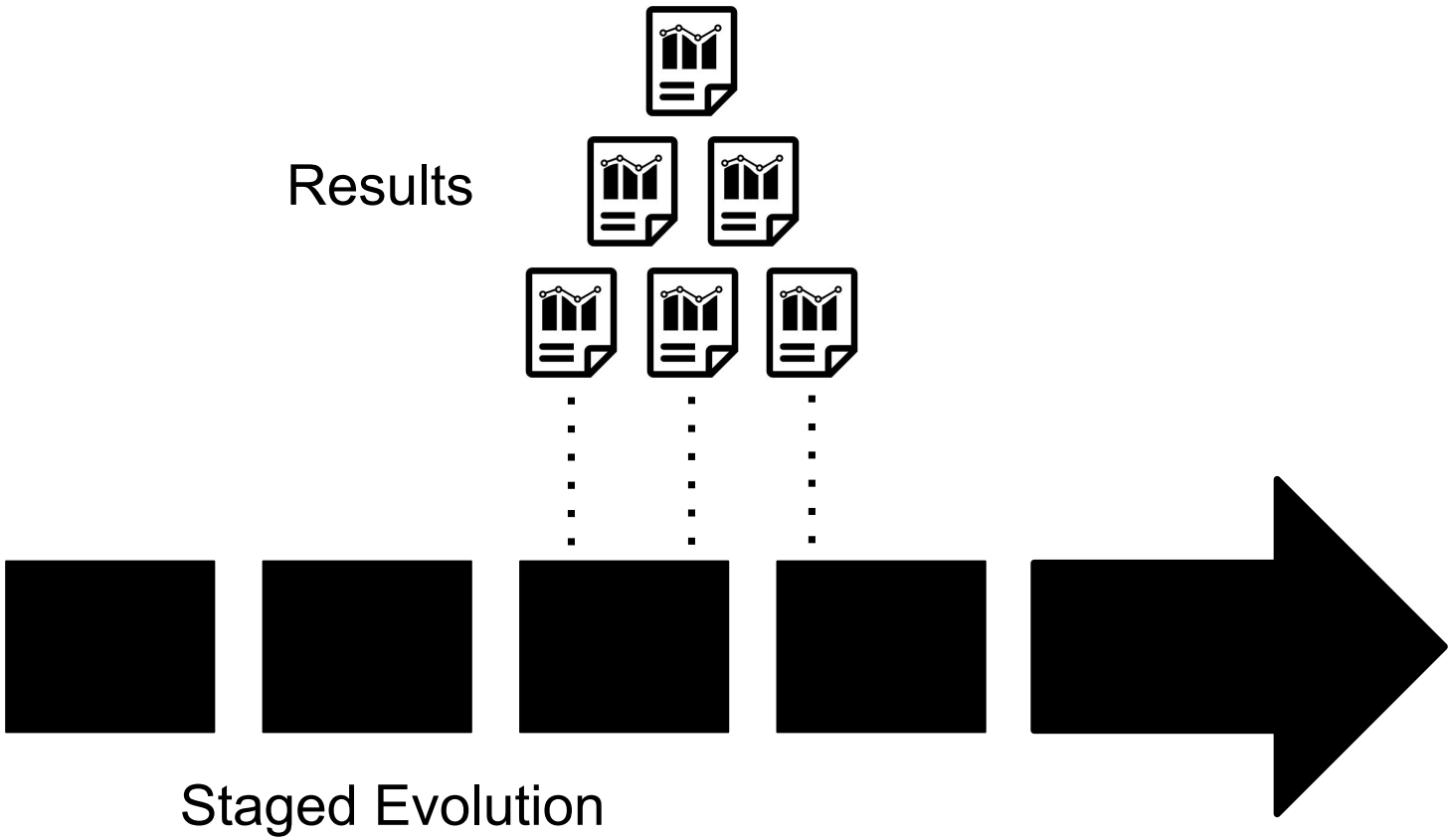


Stratified specialization across groups,

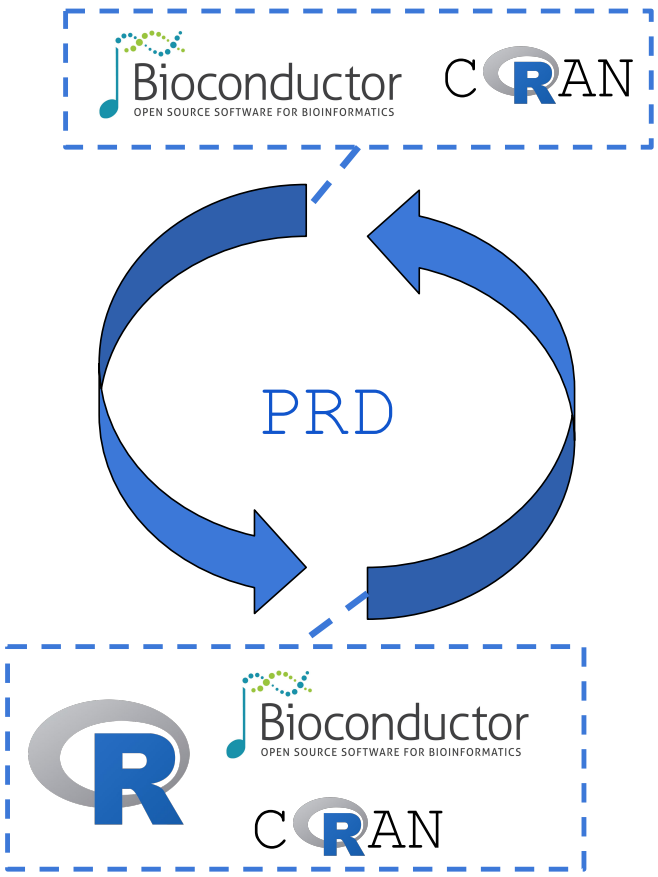
Metered evolution



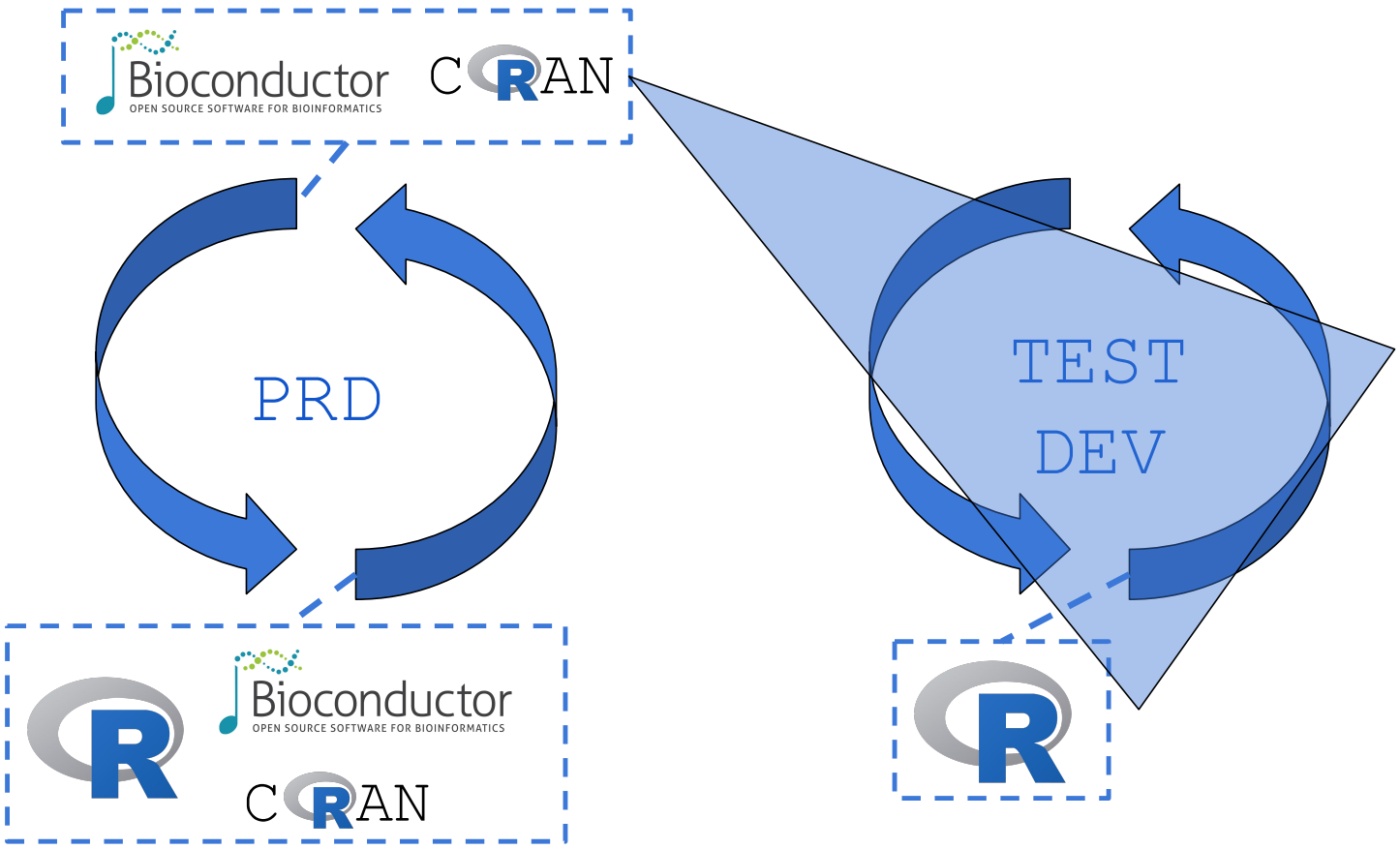
Metered evolution



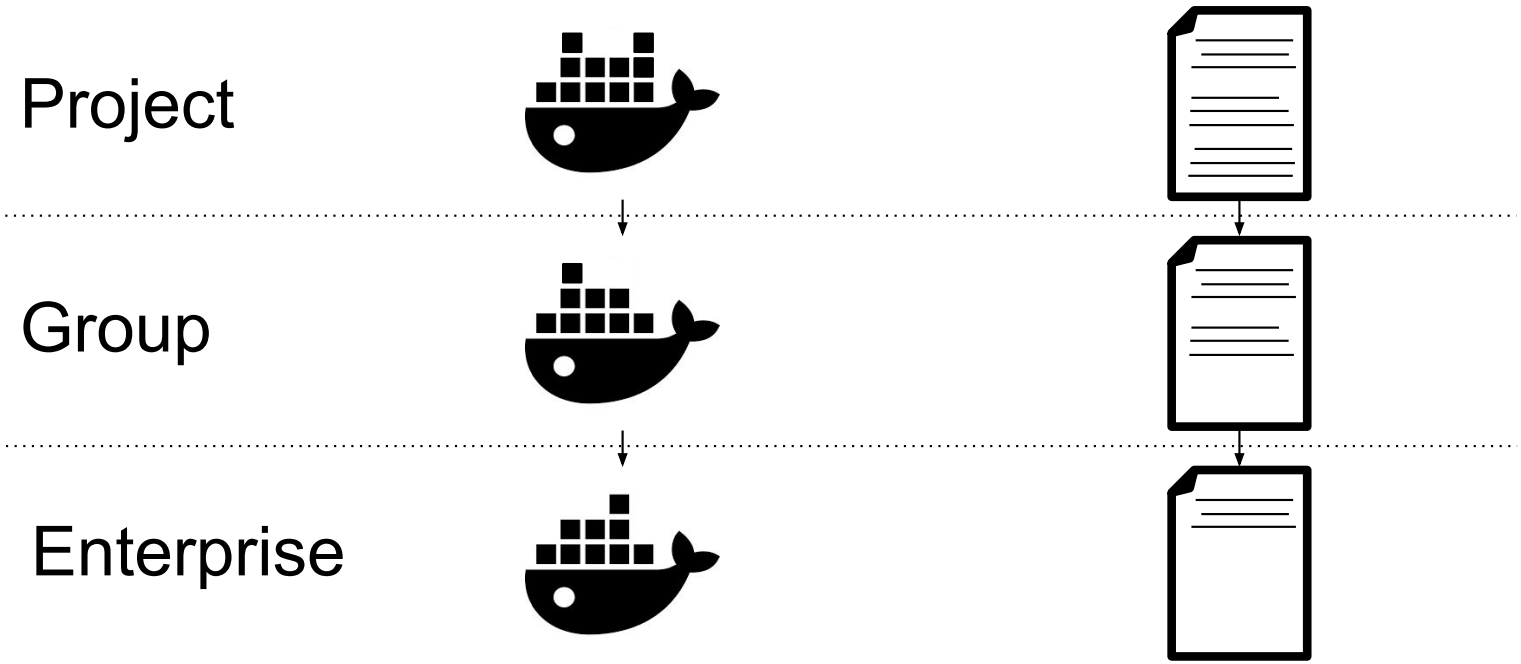
Graded evolutionary policies



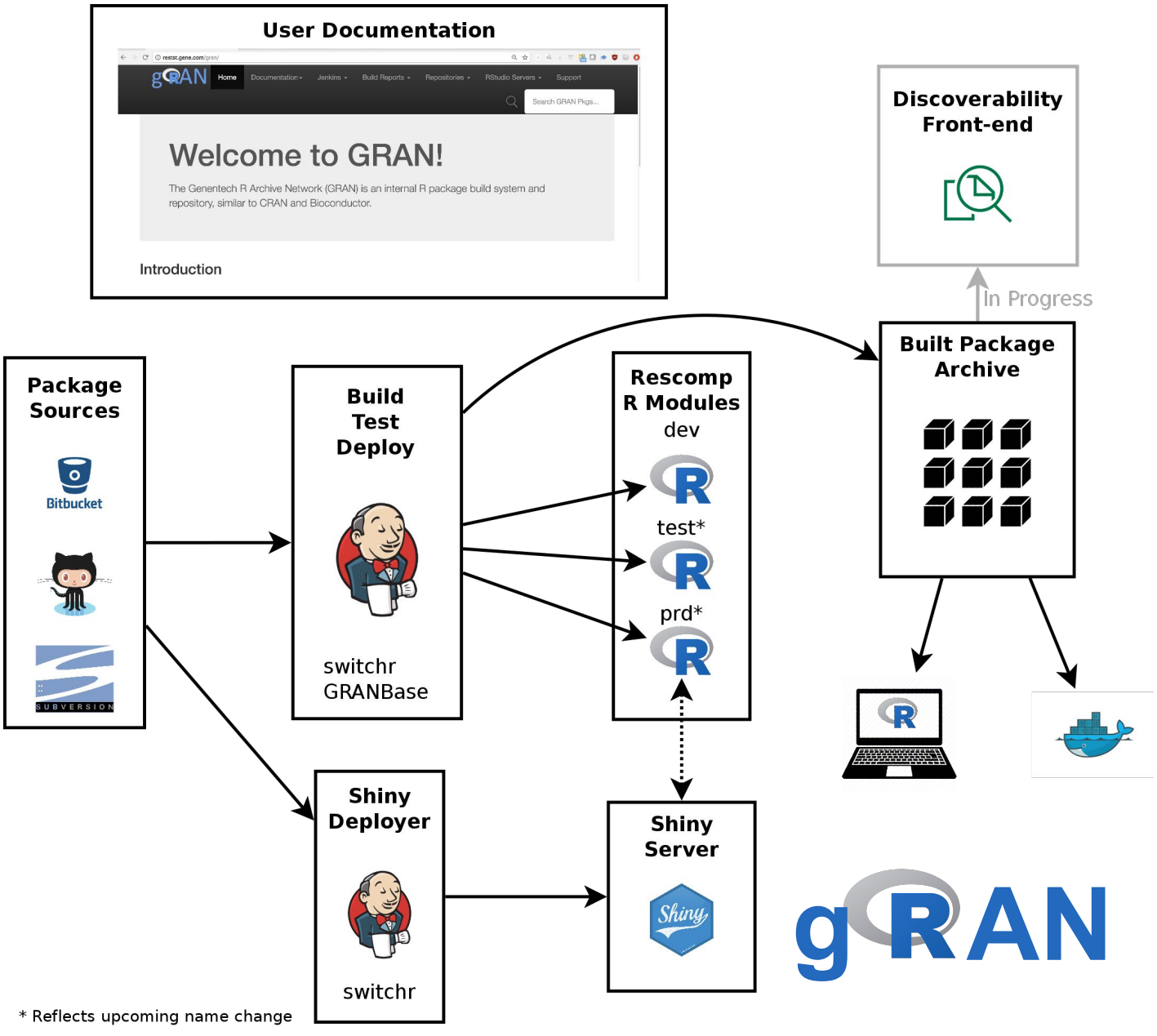
Graded evolutionary policies



Stratified specialization



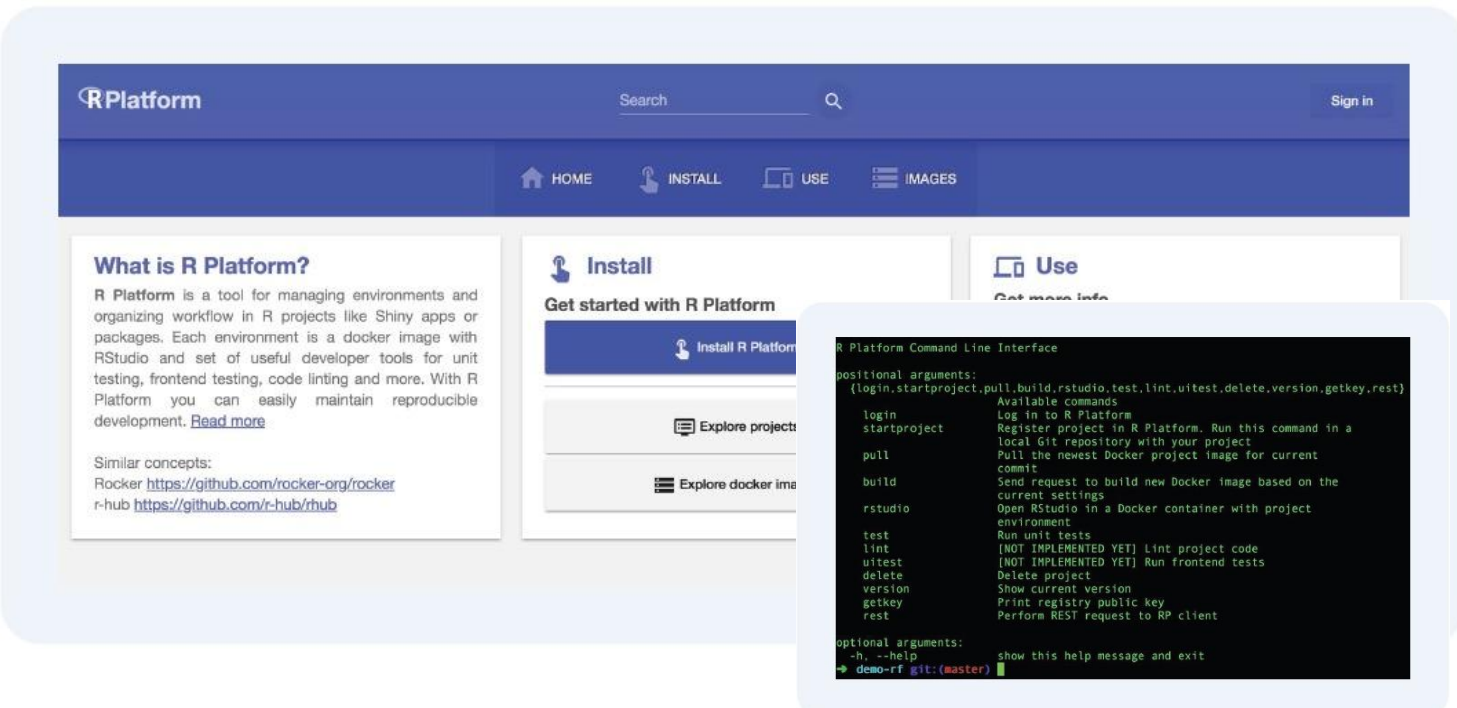
Environment orchestration (Gabe Becker)



Rapid deployment with the *R Platform*

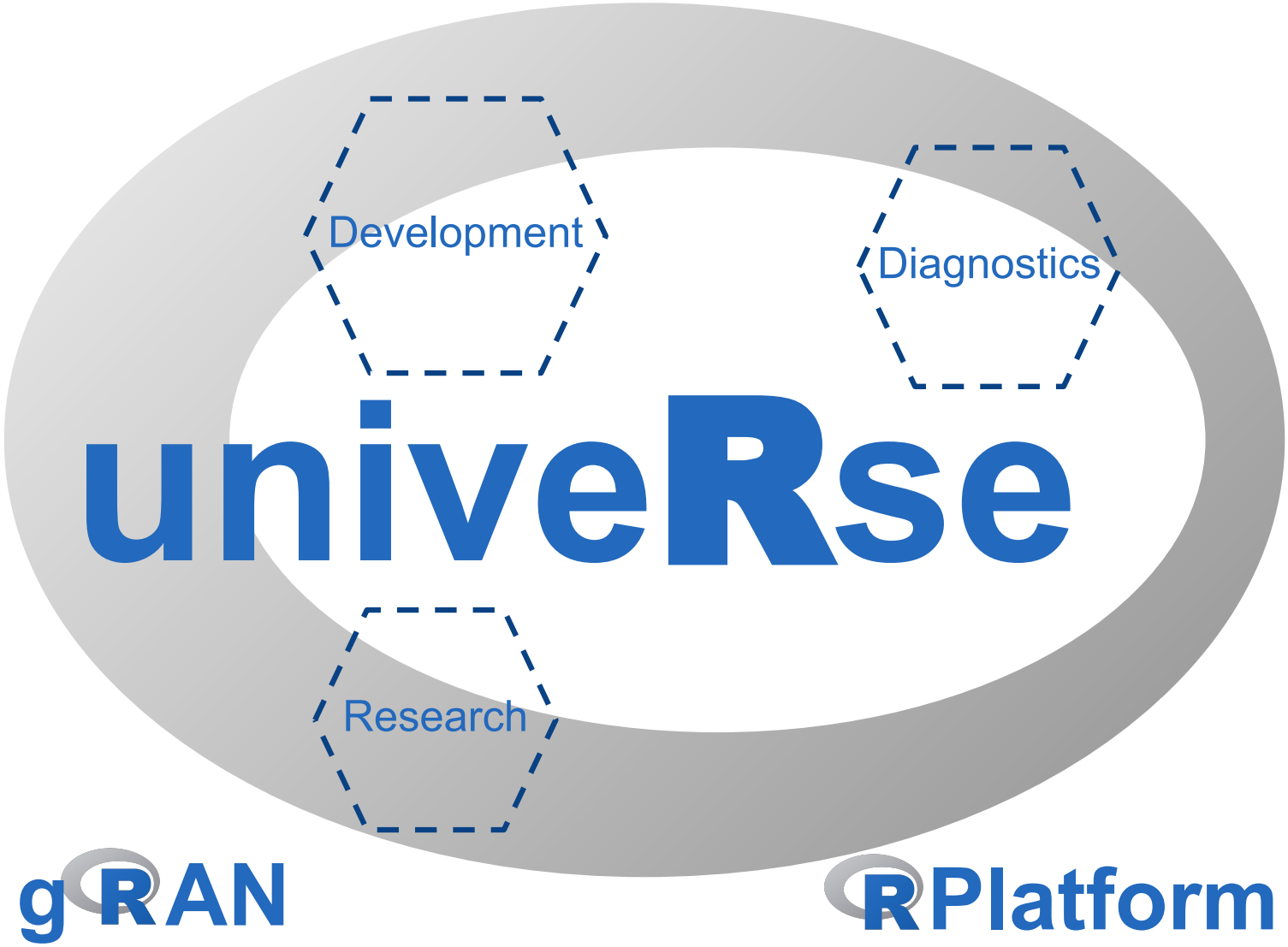
Dariusz Ratman, et al.

- The R Platform provides agile deployment of project-specific development and analytical environments through containers
- Automatic generation of Docker containers from R packages
- Share containers through an internal registry
- Automatic continuous integration and testing in multiple environments

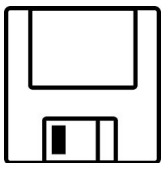


The Roche R Universe

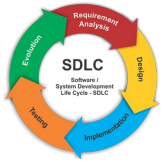
with Reinhold Koch, Sebastian Wolf, Gabe Becker



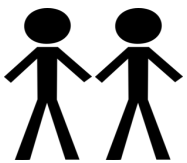
Summary



Leveraging R means becoming a software company



The software incubator translates scientist innovations into reliable software



Scientific software is challenged to enable active collaboration

gRAN

Multi-layered and strategically evolving environments can unify scientific computing across the enterprise