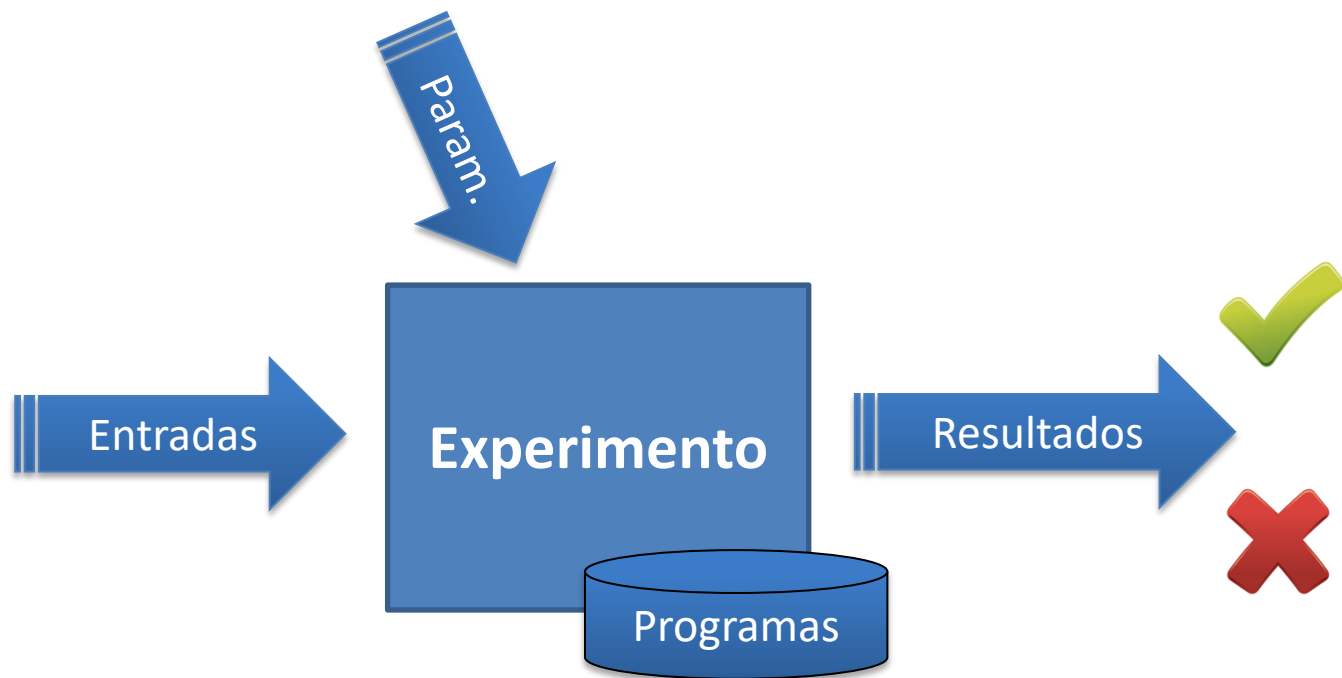


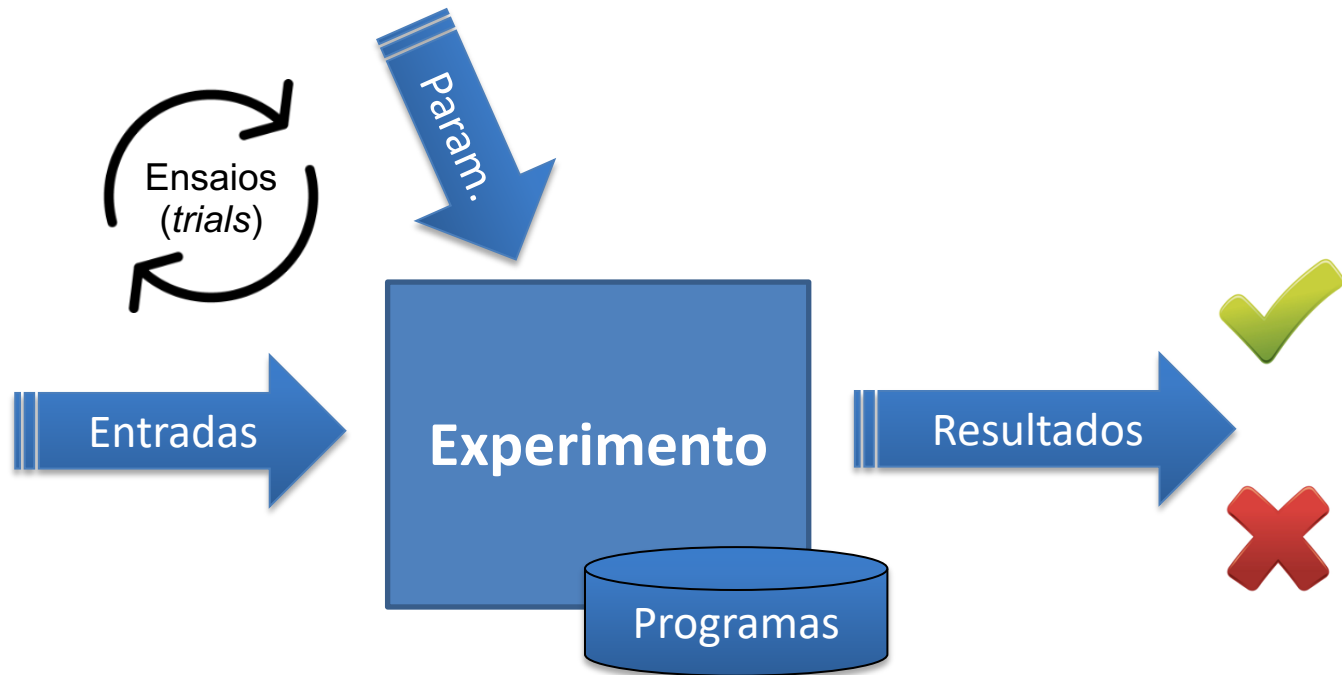
# noWorkflow: Ajudando Cientistas a Fazer Ciência



# Experimentos Científicos



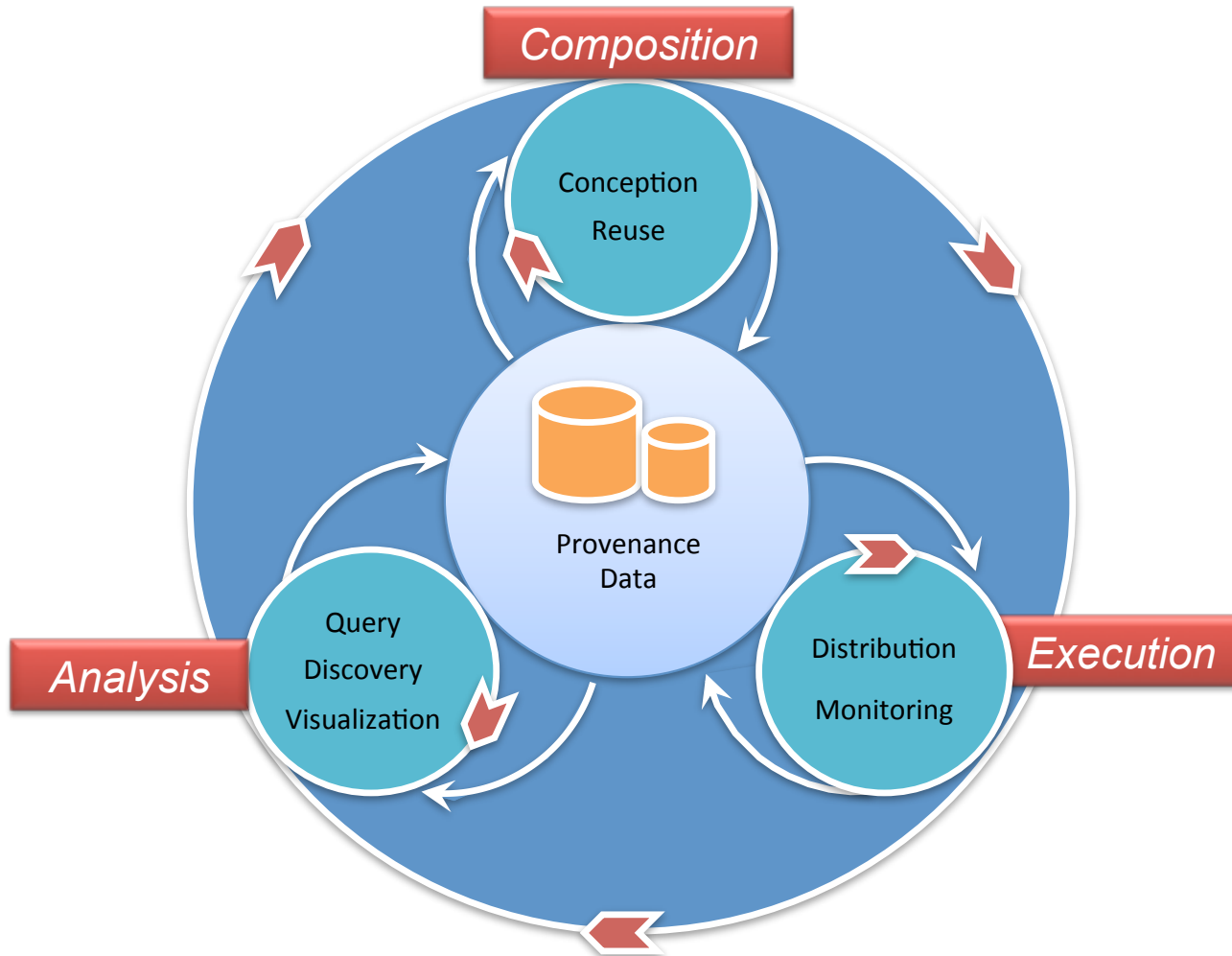
# Várias Tentativas e Repetições



# Como entender os dados?

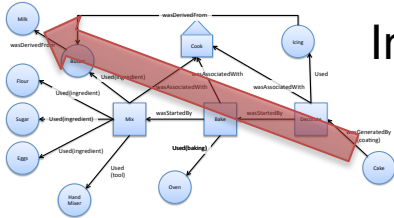


# Proveniência é a chave!

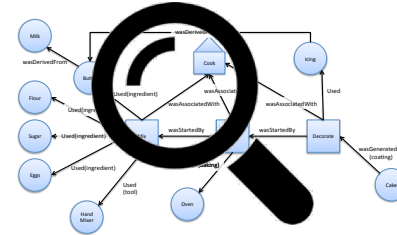


Mattoso et al. "Towards supporting the life cycle of large scale scientific experiments." IJBPM 5(1) 2010

# Proveniência é útil para...

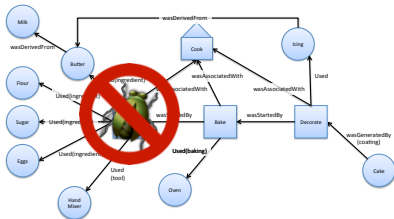
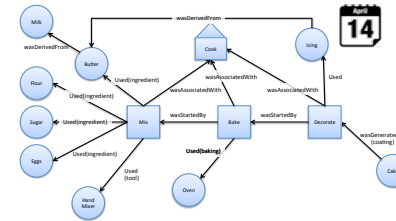
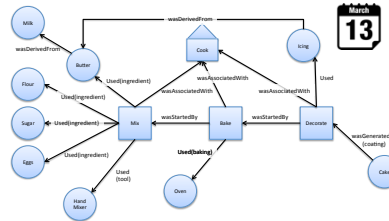


Interpretar e entender os resultados

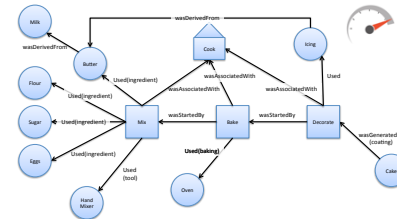


Auditoria

Reprodutibilidade



Debugging



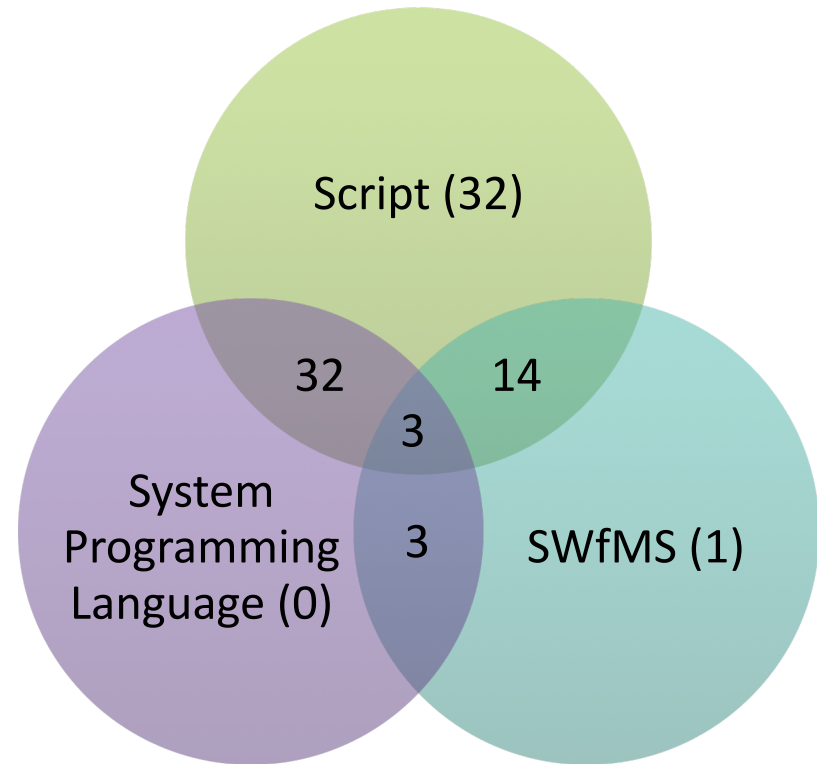
Caching

Entre outras aplicações (atribuição, reuso, compartilhamento, ...)

# Muitos Cientistas usam Scripts

95%

dos participantes\*  
citaram **scripts** entre as  
ferramentas preferidas  
ou usadas com mais  
frequência para  
executar experimentos

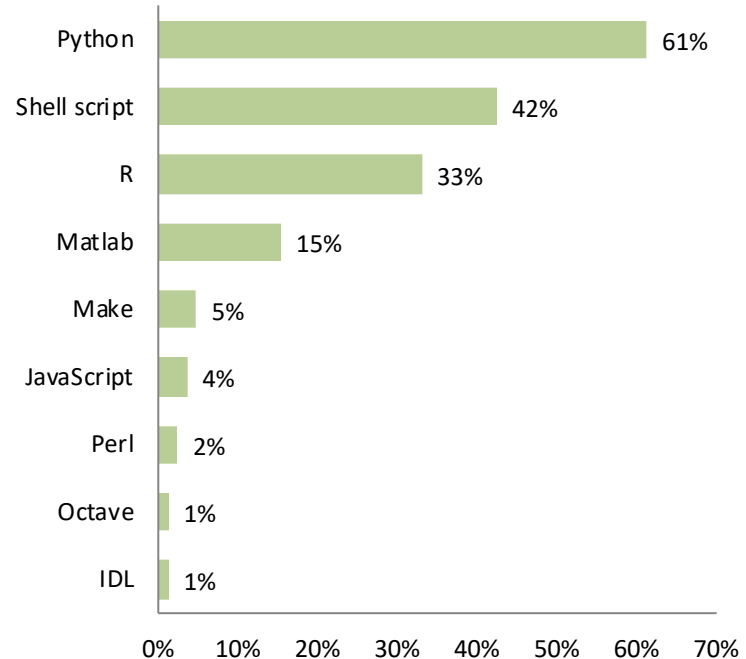


\*Survey sent to AMC@UvA (Olabarriaga), UFRJ (Mattoso), DATAONE (newsletter), DBBras (mailing list), FIOCRUZ (Davila), USP (Traina), INRIA-Montpellier (Zenith group), LNCC (Ocana), PW 2016 TPC, SciPyLA (Telegram), Software Carpentry (mailing list), U. Nantes (Gaignard), UPENN (Davidson), receiving 85 answers.

# Muitos Cientistas usam Python

61%

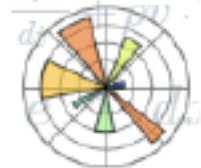
dos participantes\*  
citaram **Python** entre  
suas ferramentas  
preferidas ou mais  
usadas para executar  
experimentos



\*Survey sent to AMC@UvA (Olabarriaga), UFRJ (Mattoso), DATAONE (newsletter), DBBras (mailing list), FIOCRUZ (Davila), USP (Traina), INRIA-Montpellier (Zenith group), LNCC (Ocana), PW 2016 TPC, SciPyLA (Telegram), Software Carpentry (mailing list), U. Nantes (Gaignard), UPENN (Davidson), receiving 85 answers.



# Iniciativas Baseadas em Scripts para a Ciência não Capturam Proveniência



# Recentemente, surgiram algumas ferramentas

Astro-WISE	Becker and Chambers	Bochner, Gude, and Schreiber	CPL	CXXR	Datatrack
ES3	ESSW	IncPy	Lancet	Magni	Michaelides et al
noWorkflow	Provenance Curious	pypet	RDataTracker	Sacred	SisGExp
StarFlow	Sumatra	Tariq, Ali, and Gehani	Variolite	VCR	versuchung
	WISE	YesWorkflow	YW*NW		

# noWorkflow

- **Coleta proveniência** de scripts **Python** de forma **transparente**
- Considera múltiplas execuções (**ensaios**)
- Fornece **visualizações** para análise de proveniência
- Permite **consultar** proveniência in diferentes linguagens (SQL e Prolog)

# Instalação e Execução

Ao invés de rodar

```
$ python experiment.py
```

Instalar o noWorkflow (apenas uma vez)

```
$ pip install noworkflow[all]
```

E executar

```
$ now run experiment.py
```

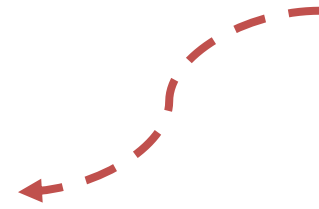
# Executando...

```
$ now run -v experiment.py
[now] removing noWorkflow boilerplate
[now] setting up local provenance store
[now] collecting definition provenance
[now] registering user-defined functions
[now] collecting deployment provenance
[now] registering environment attributes
[now] searching for module dependencies
[now] registering provenance from 703 modules
[now] collecting execution provenance
[now] executing the script
[now] the execution of trial 1 finished successfully
```

# Executando...

```
$ now run -v experiment.py  
[now] removing noWorkflow boilerplate  
[now] setting up local provenance store  
[now] collecting definition provenance  
[now] registering user-defined functions  
[now] collecting deployment provenance  
[now] registering environment attributes  
[now] searching for module dependencies  
[now] registering provenance from 703 modules  
[now] collecting execution provenance  
[now] executing the script  
[now] the execution of trial 1 finished successfully
```

*Código do script*



# Executando...

```
$ now run -v experiment.py
[now] removing noWorkflow boilerplate
[now] setting up local provenance store
[now] collecting definition provenance
[now] registering user-defined functions
[now] collecting deployment provenance
[now] registering environment attributes
[now] searching for module dependencies
[now] registering provenance from 703 modules
[now] collecting execution provenance
[now] executing the script
[now] the execution of trial 1 finished successfully
```

*Código do script*

*Ambiente  
(módulos,  
versões, SO,  
etc.)*

# Executando...

```
$ now run -v experiment.py
[now] removing noWorkflow boilerplate
[now] setting up local provenance store
[now] collecting definition provenance
[now] registering user-defined functions
[now] collecting deployment provenance
[now] registering environment attributes
[now] searching for module dependencies
[now] registering provenance from 703 modules
[now] collecting execution provenance
[now] executing the script
[now] the execution of trial 1 finished successfully
```

*Código do script*

*Ambiente  
(módulos,  
versões, SO,  
etc.)*

*Início, fim,  
Valores de  
variáveis, etc.*



# Executando...

```
$ now run -v experiment.py
[now] removing noWorkflow boilerplate
[now] setting up local provenance store
[now] collecting definition provenance
[now] registering user-defined functions
[now] collecting deployment provenance
[now] registering environment attributes
[now] searching for module dependencies
[now] registering provenance from 703 modules
[now] collecting execution provenance
[now] executing the script
[now] the execution of trial 1 finished successfully
```

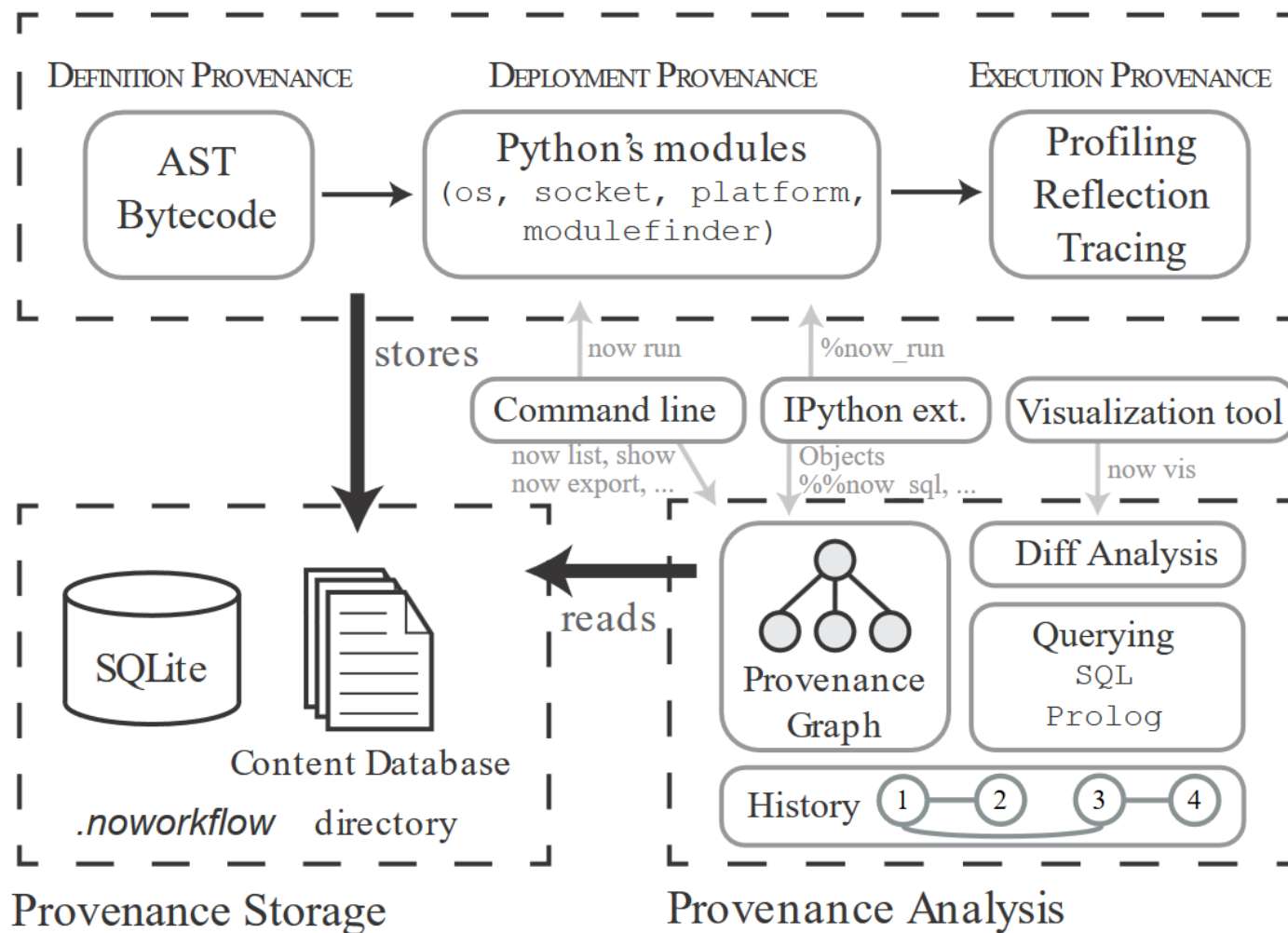
*Código do script*

*Ambiente  
(módulos,  
versões, SO,  
etc.)*

*Início, fim,  
Valores de  
variáveis, etc.*

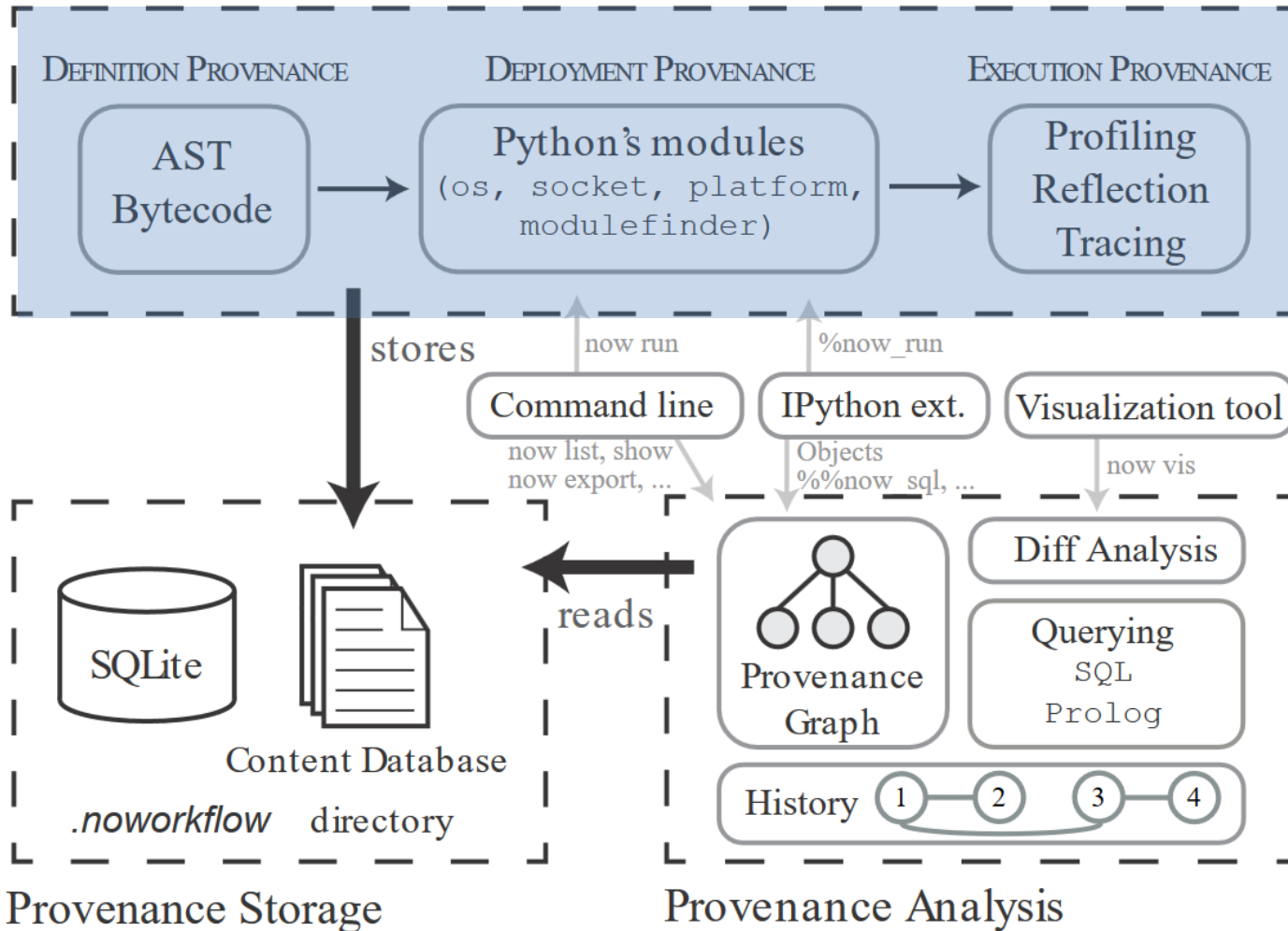
# Arquitetura

## Provenance Collection



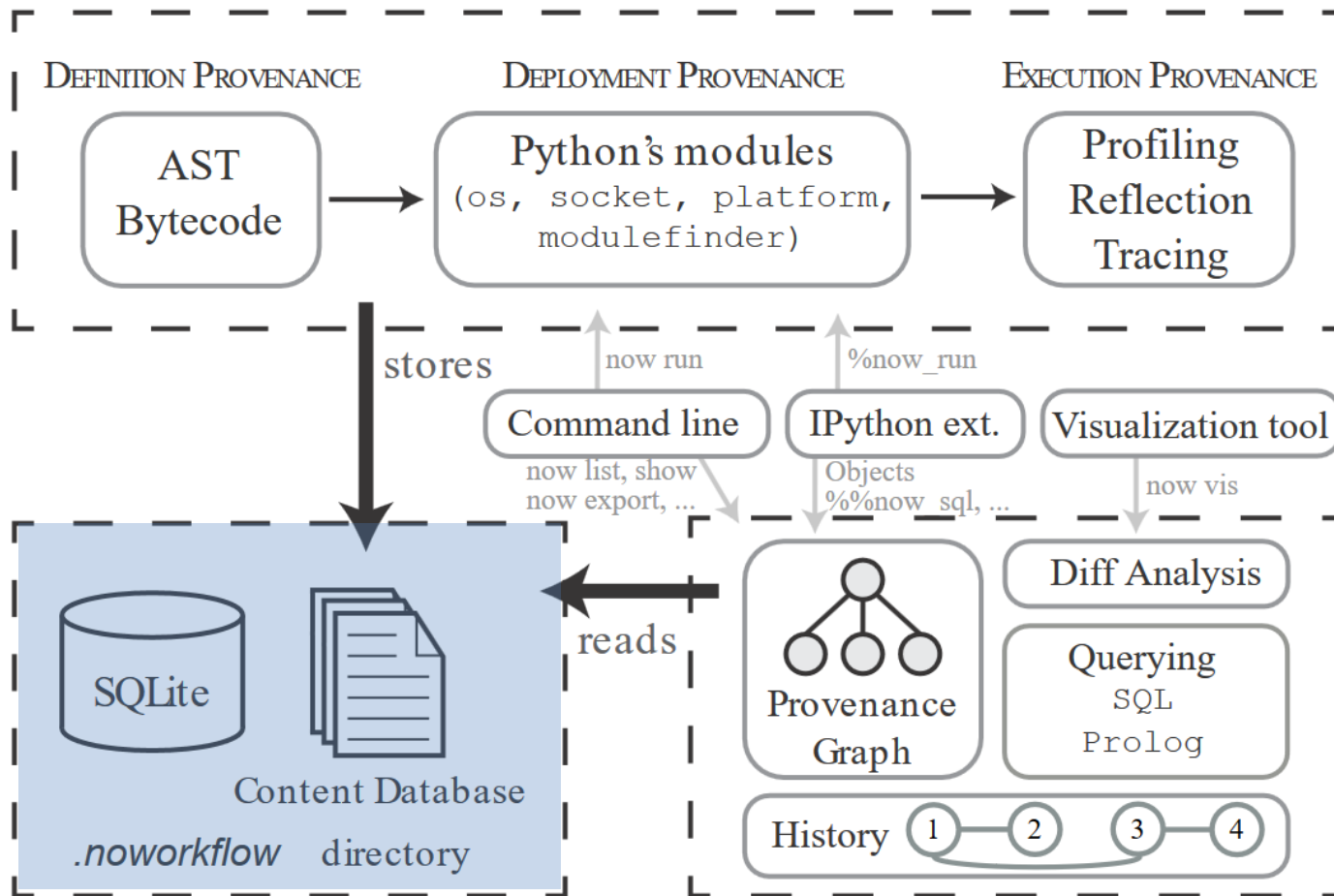
# Arquitetura

## Provenance Collection



# Arquitetura

## Provenance Collection

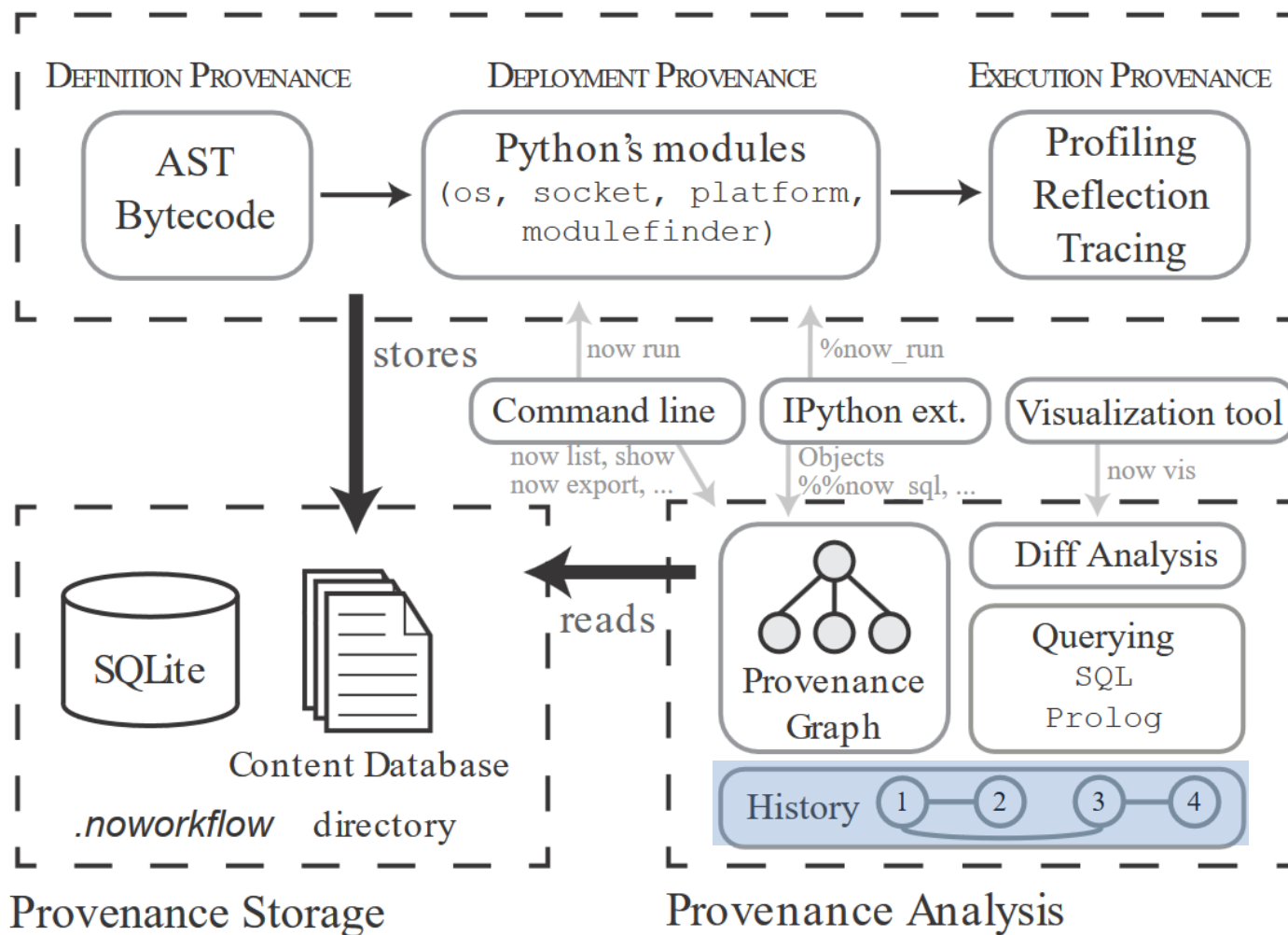


Provenance Storage

Provenance Analysis

# Arquitetura

## Provenance Collection



# Histórico de Ensaios

trial 1



Cada nó no grafo é representado por um número de versão **Major.Minor.Patch**:

- **Patch**: reexecução com mesmo parâmetros e mesmo código.
- **Minor**: mesmo código, mas parâmetros diferentes.
- **Major**: código diferente

# Histórico de Ensaios

## Modificações em experiment.py

```
$ now run experiment.py
```



# Histórico de Ensaios

Modificações em `experiment.py`, sem execução

Restauração do trial 1:

```
$ now restore 1.1.1 (ou $ now restore 1)
```

Execução do trial 4 (trial 3 é um trial de backup)

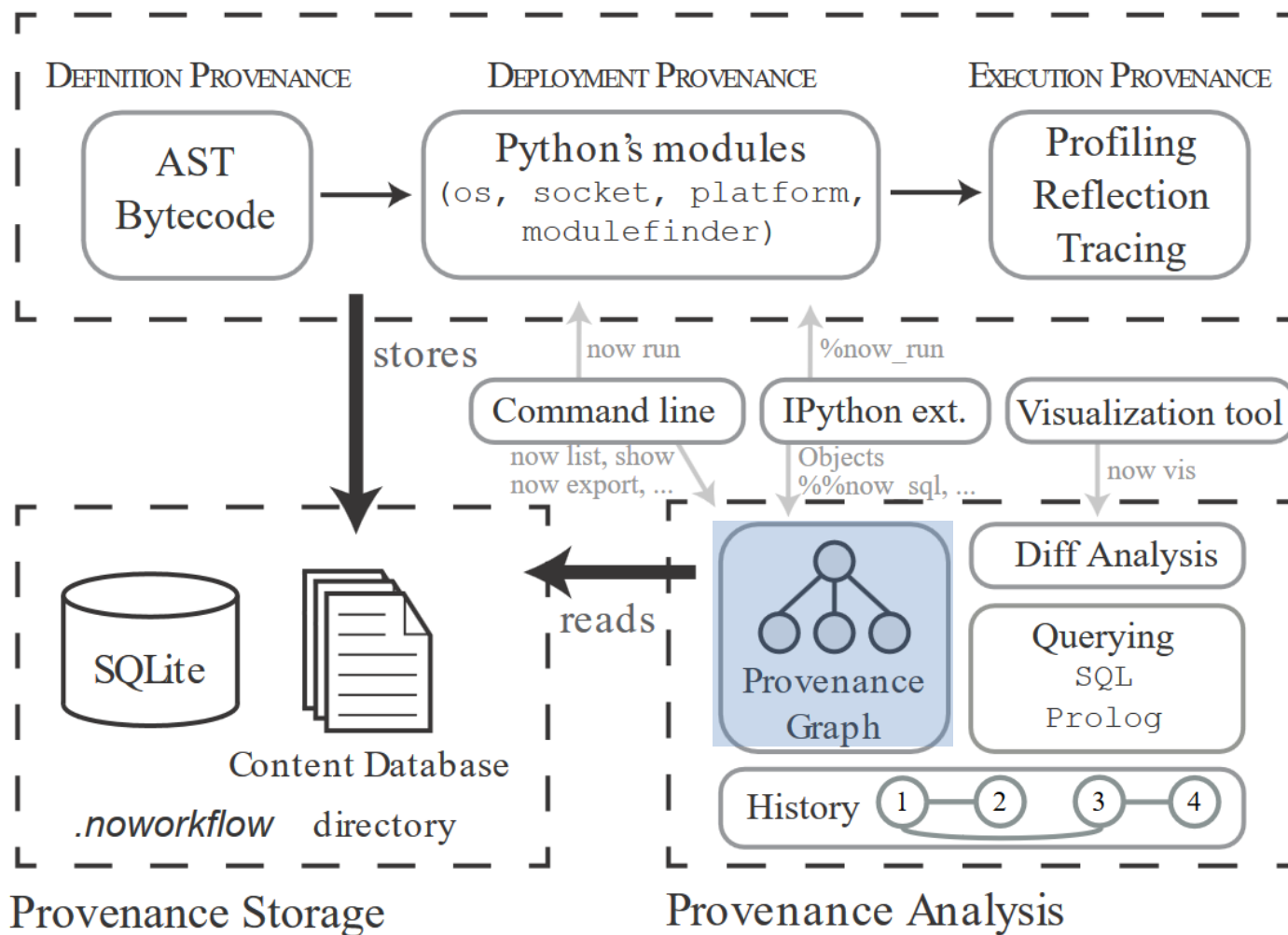
```
$ now run experiment.py
```



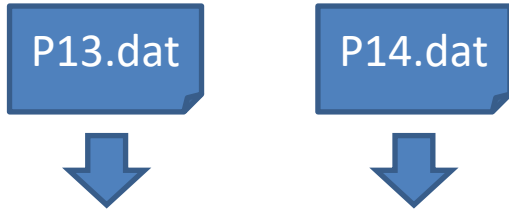


# Arquitetura

## Provenance Collection



# Resultado e Proveniência



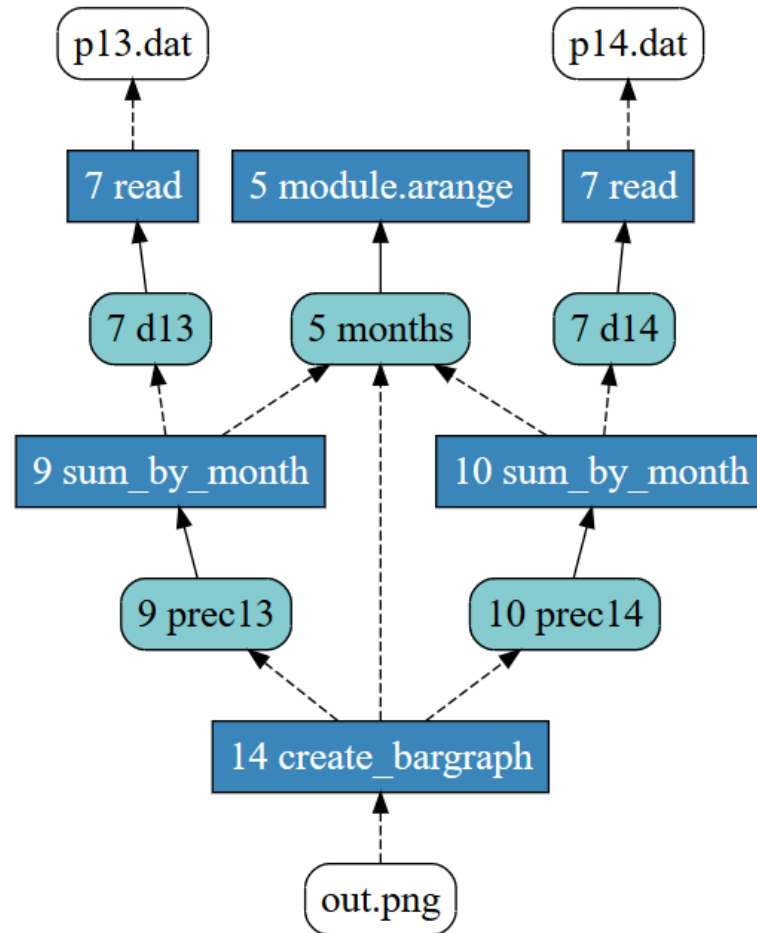
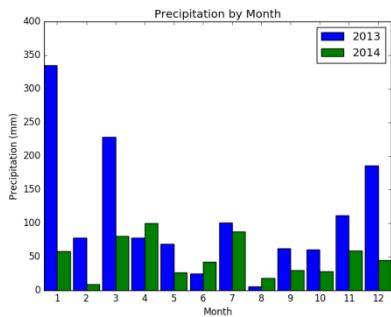
```
import numpy as np
from precipitation import read, sum_by_month
from precipitation import create_bargraph

months = np.arange(12) + 1

d13, d14 = read("p13.dat"), read("p14.dat")

prec13 = sum_by_month(d13, months)
prec14 = sum_by_month(d14, months)

create_bargraph("out.png", months,
                ["2013", "2014"],
                prec13, prec14)
```



# DEMONSTRAÇÃO

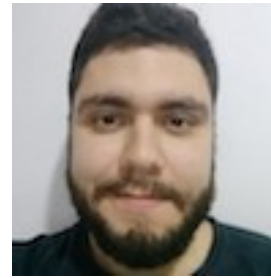
# Equipe noWorkflow



João Felipe Pimentel  
IC/UFF



Eduardo Jandre  
IC/UFF



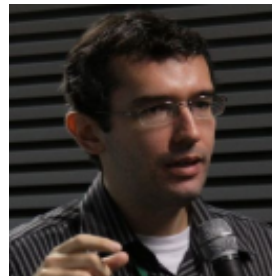
Vynicius Pontes  
IC/UFF



Vitor Lemos  
IC/UFF



Vanessa Braganholo  
IC/UFF



Leonardo Murta  
IC/UFF



Juliana Freire  
NYU

# Outros Colaboradores

- Bertram Ludäscher (U. Illinois at Urbana-Champaign)
- David Koop (UMass Dartmouth)
- Fernando Chirigati (NYU)
- Khalid Belhajjame (U. Paris Dauphine)
- Paolo Missier (Newcastle U.)
- Saumen Dey (UC Davis)
- Timothy McPhillips (U. Illinois at Urbana-Champaign)

# Financiamento



# Em andamento...

- Apoio à Colaboração
- Re-execução acelerada (cache)

# noWorkflow: Ajudando Cientistas a Fazer Ciência

Experimente  
o noWorkflow!

