Parameter Server on Flink

an approach for model-parallel machine learning

27/09/2018

Distributed Computing and Analytics Workshop

Dániel Berecz
bdaniel@info.ilab.sztaki.hu

MTA SZTAKI

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 688191.

About us

- Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI)
- Data Science group
- Strong industry ties
  - Ericsson, Bosch, Portugal Telekom, etc.

MTA SZTAKI

STREAMLINE.

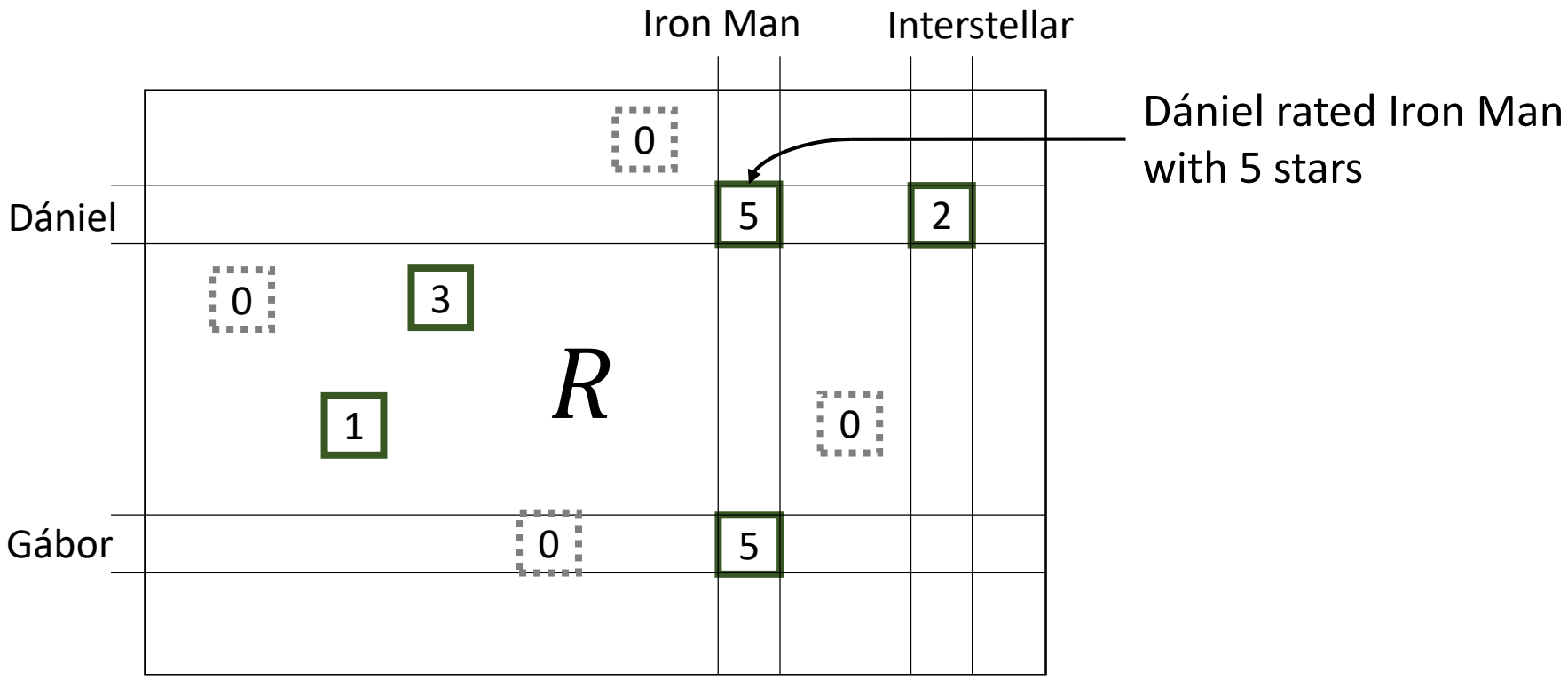
MTA SZTAKI

Agenda

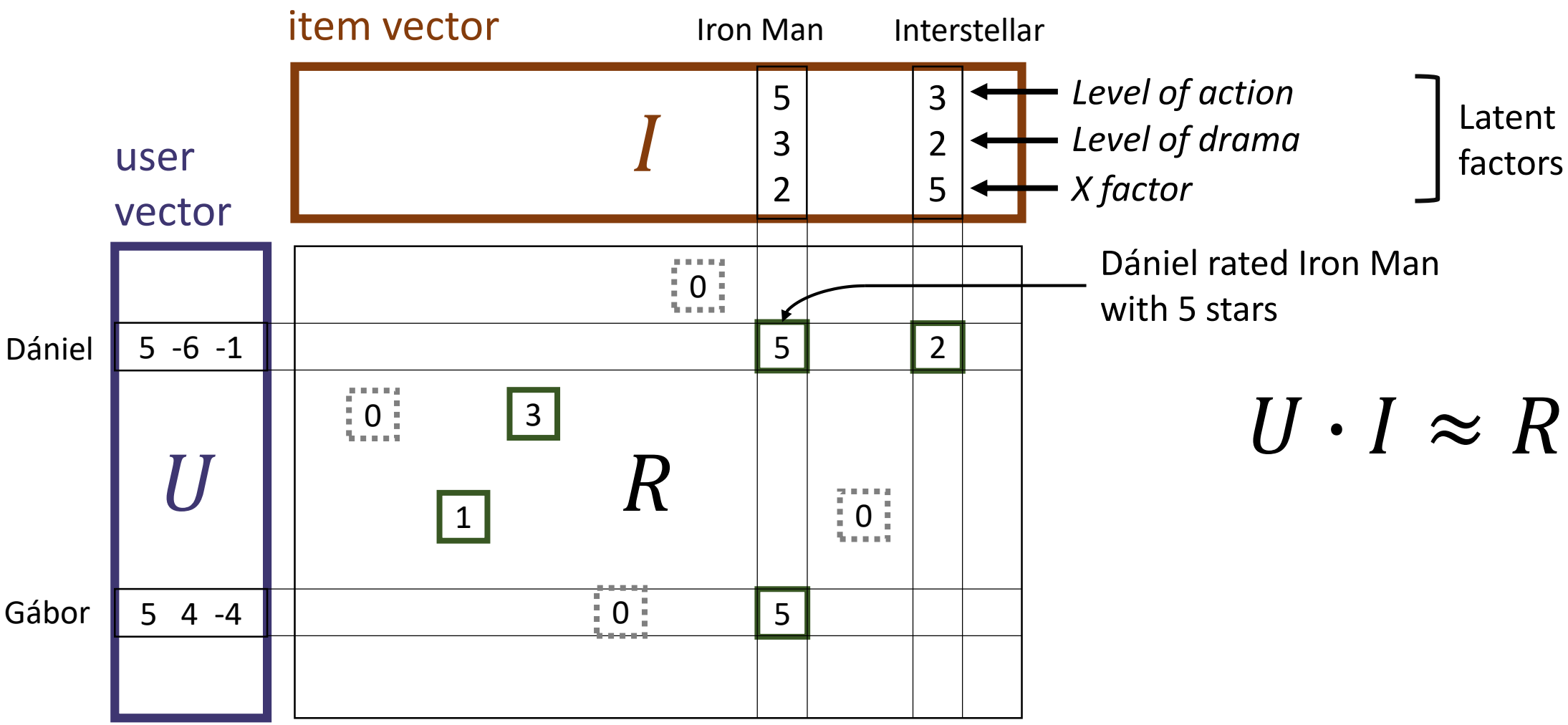
1. Model-parallel training
2. Parameter Server on Flink Streaming

Model-parallel training

Recommendation with matrix factorization

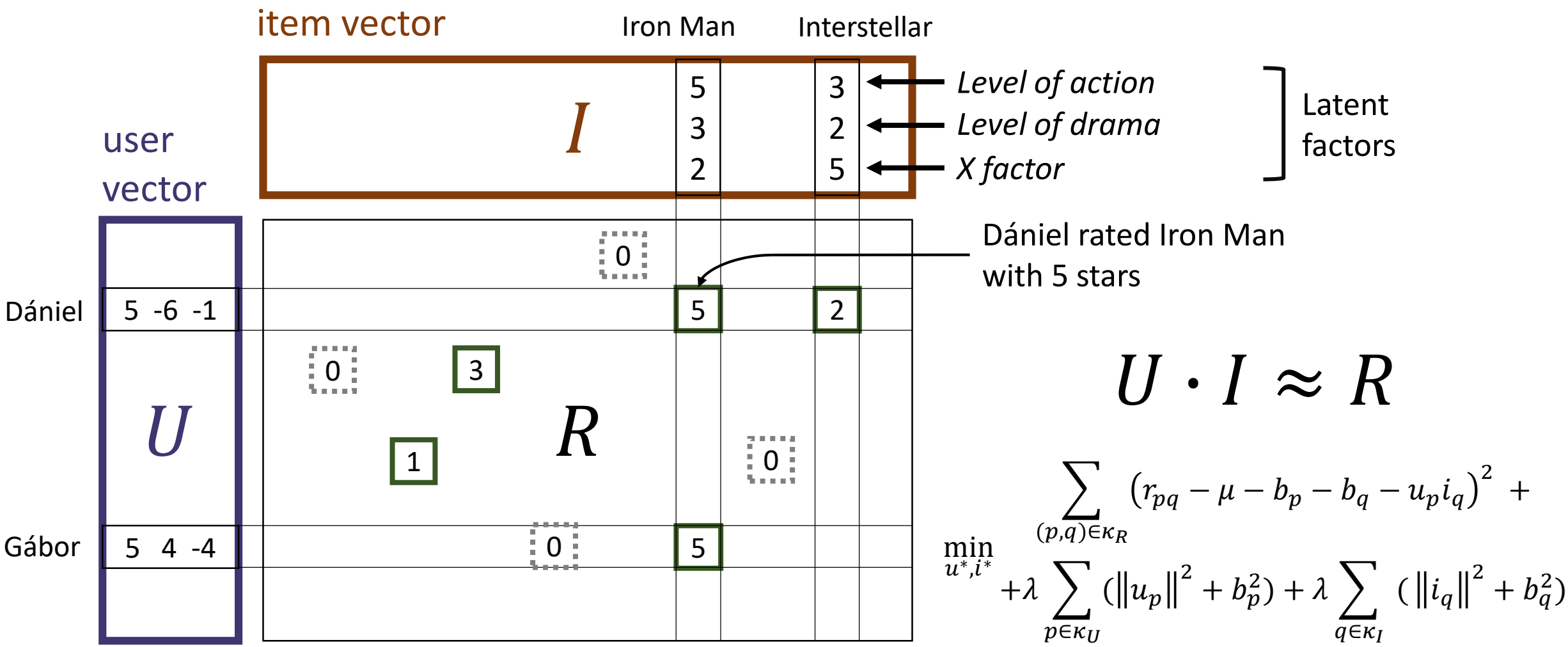


Recommendation with matrix factorization



$$U \cdot I \approx R$$

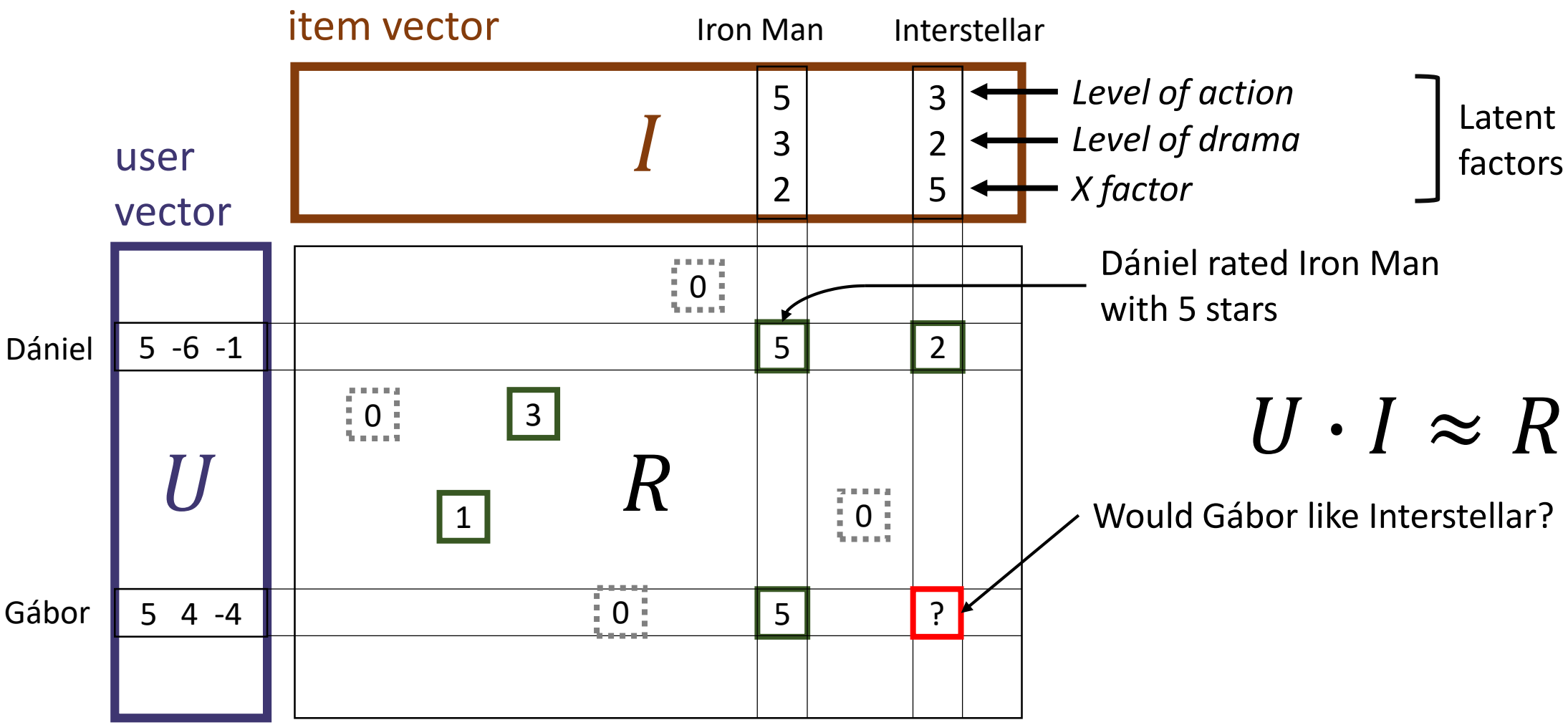
Recommendation with matrix factorization



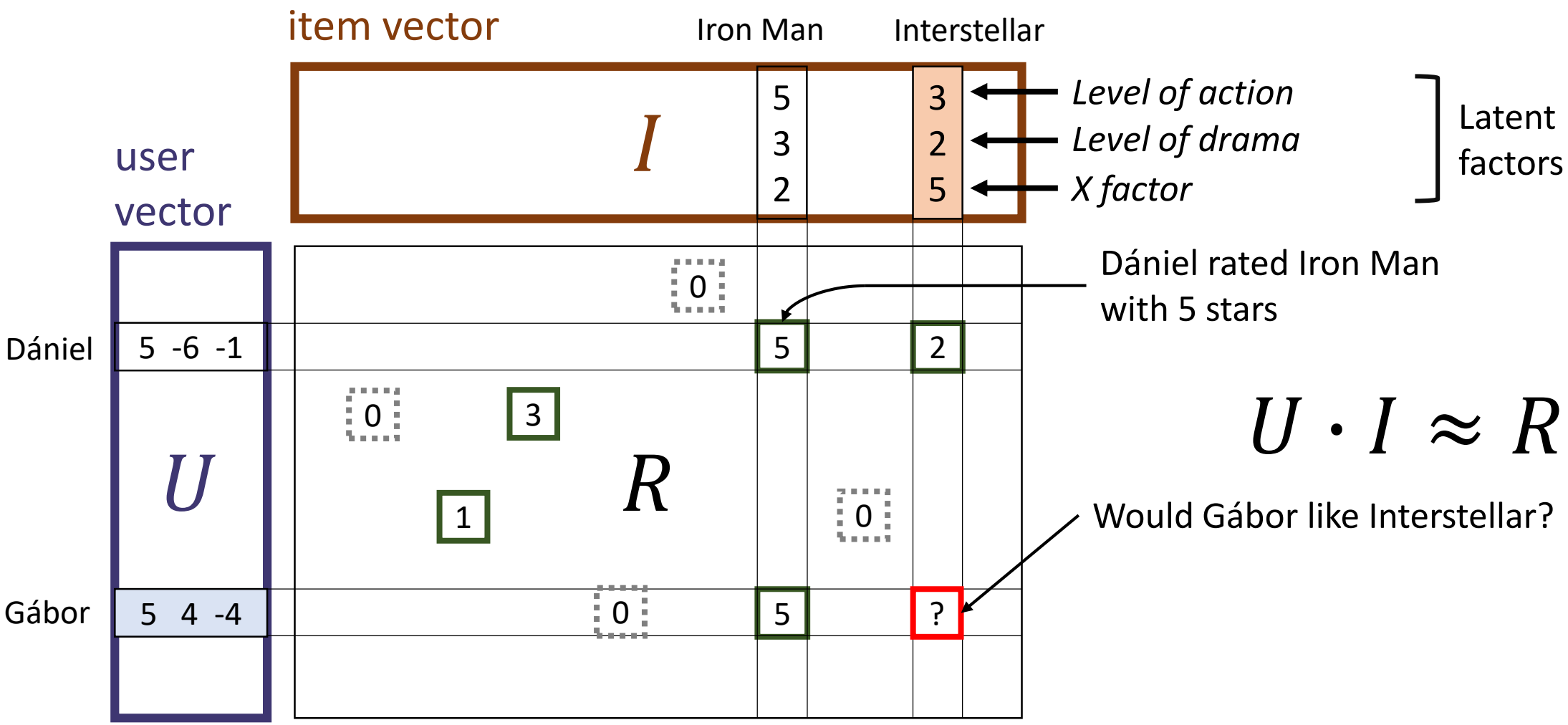
$$U \cdot I \approx R$$

$$\min_{u^*,i^*} \sum_{(p,q)\in\kappa_R} \left(r_{pq} - \mu - b_p - b_q - u_p i_q\right)^2 + \lambda \sum_{p\in\kappa_U} \left(\|u_p\|^2 + b_p^2\right) + \lambda \sum_{q\in\kappa_I} \left(\|i_q\|^2 + b_q^2\right)$$

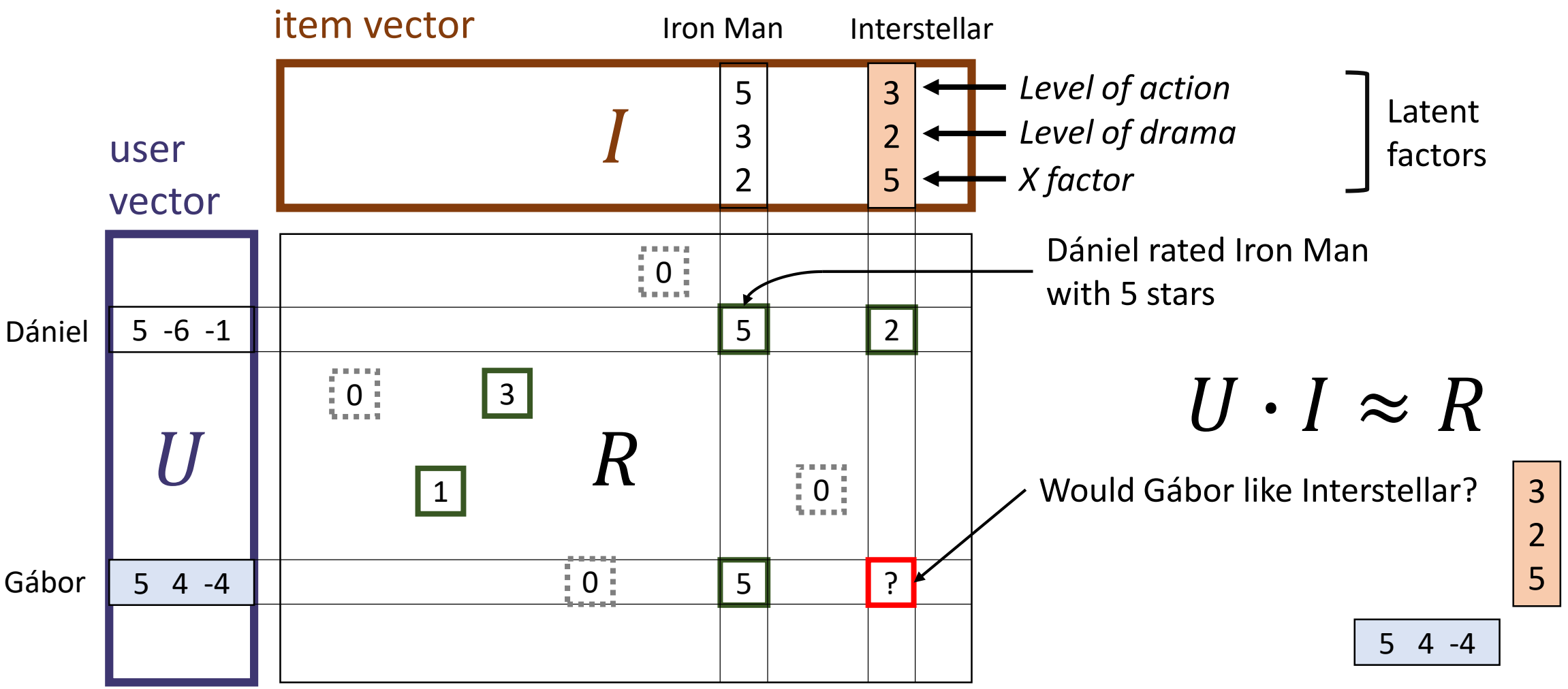
Recommendation with matrix factorization



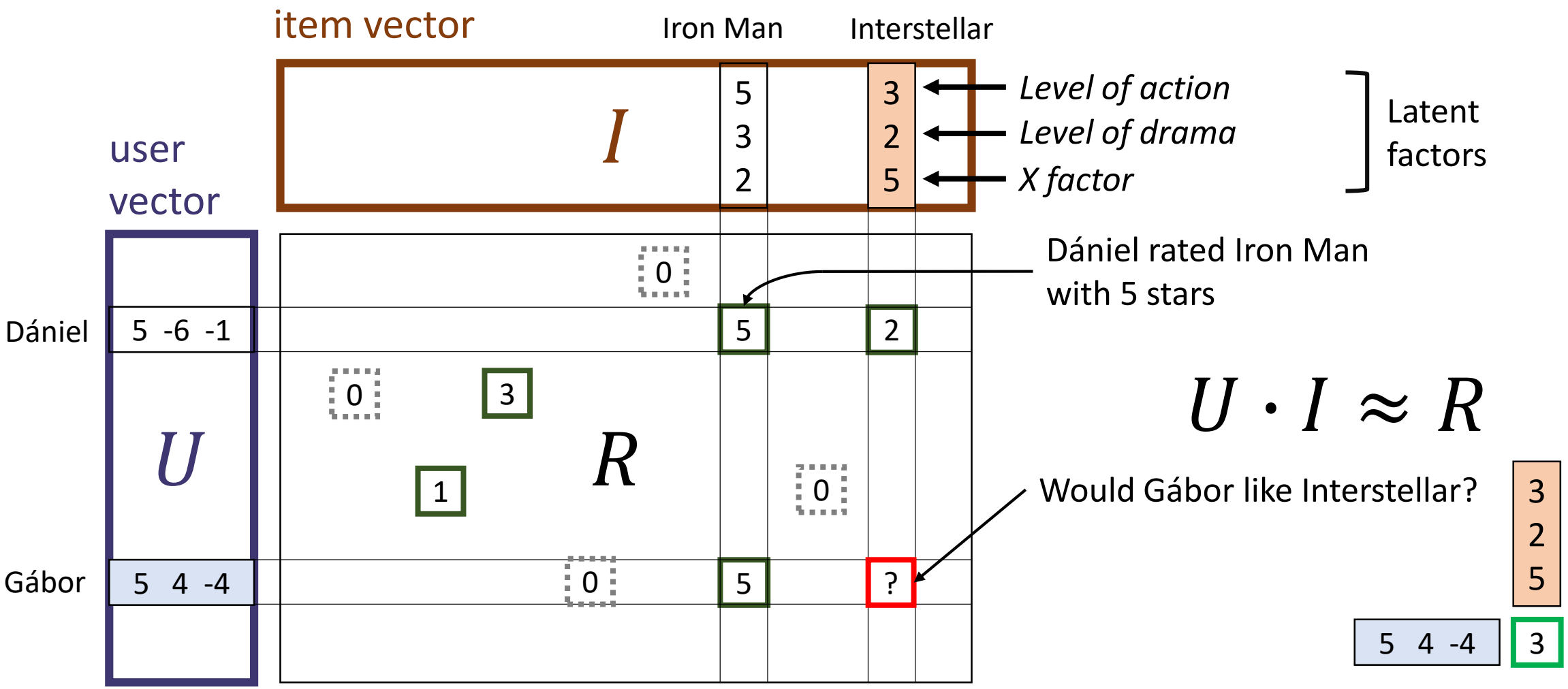
Recommendation with matrix factorization



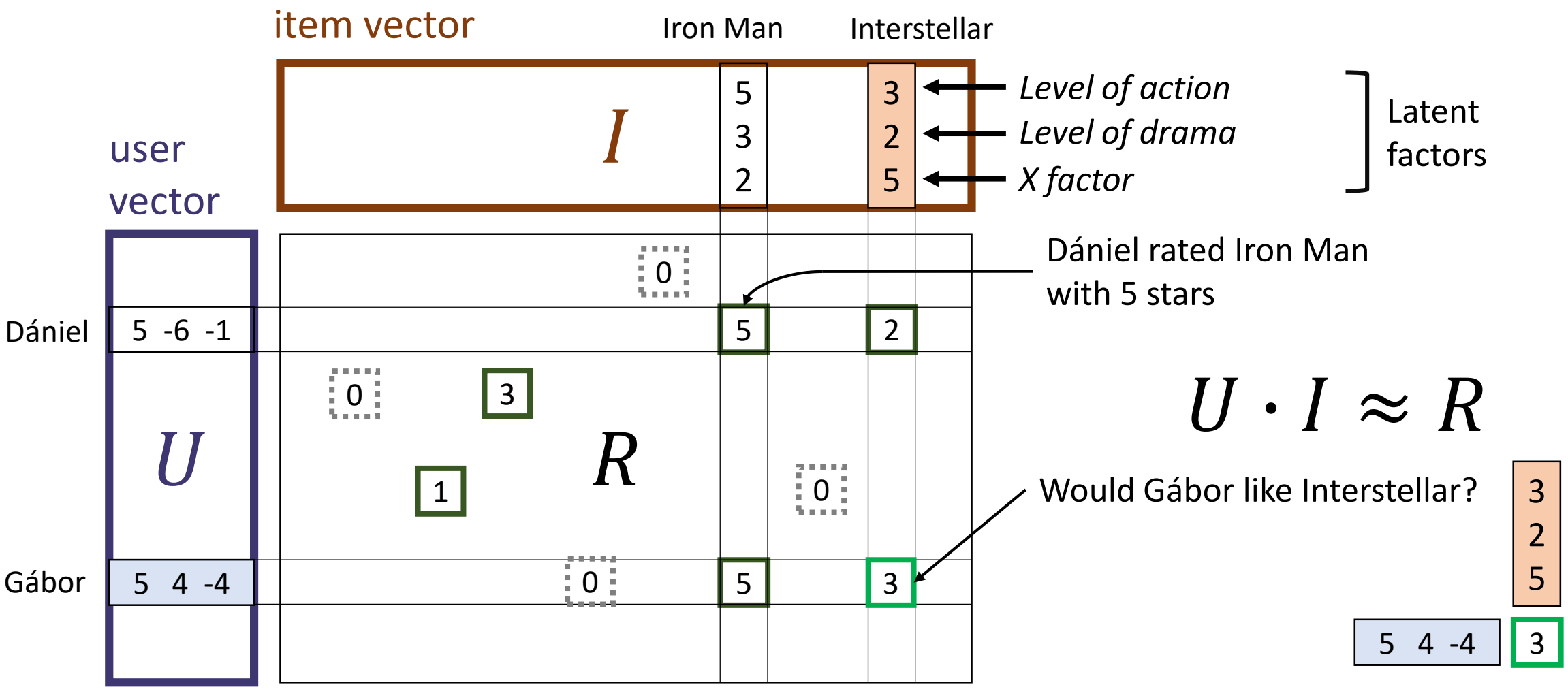
Recommendation with matrix factorization



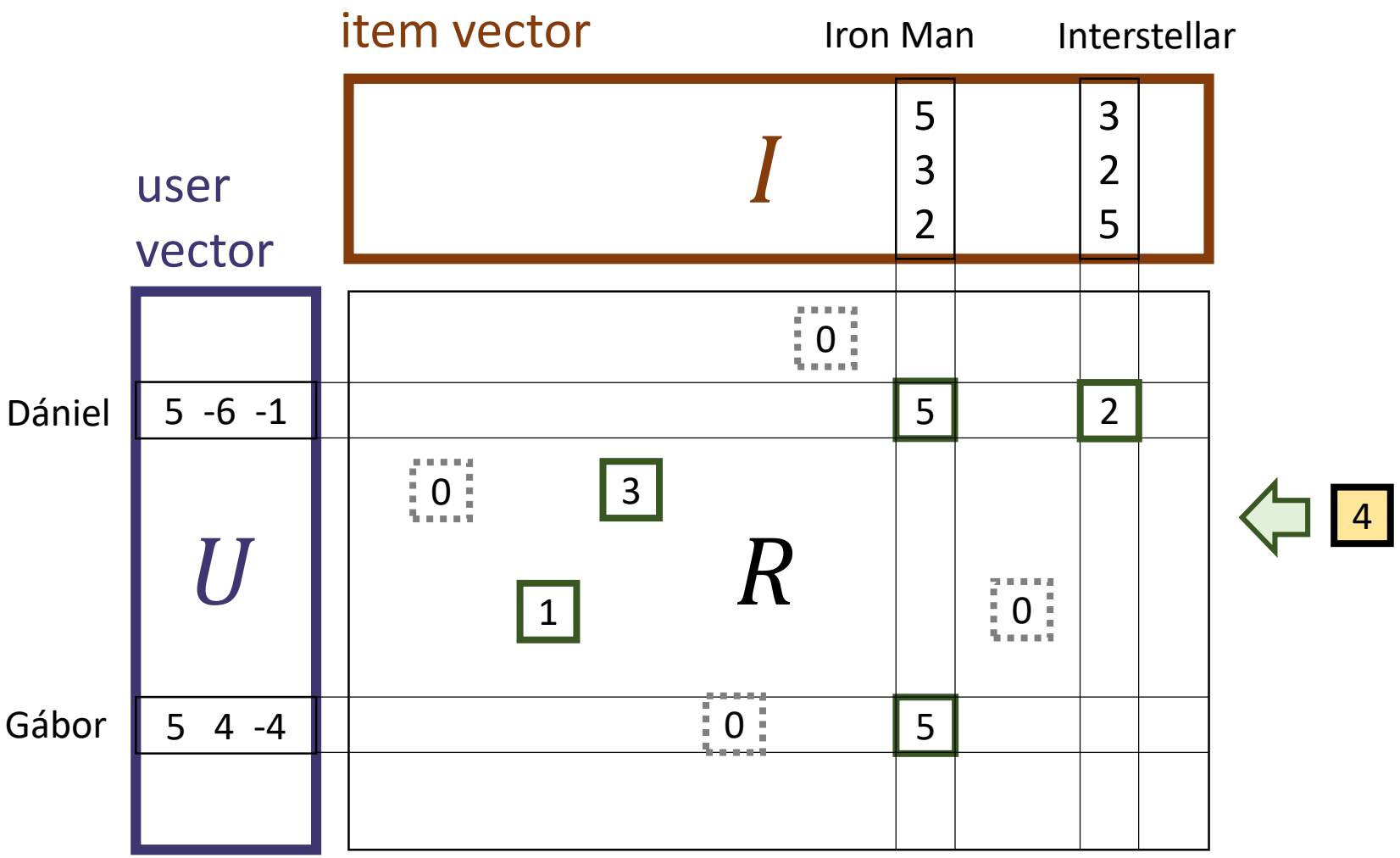
Recommendation with matrix factorization



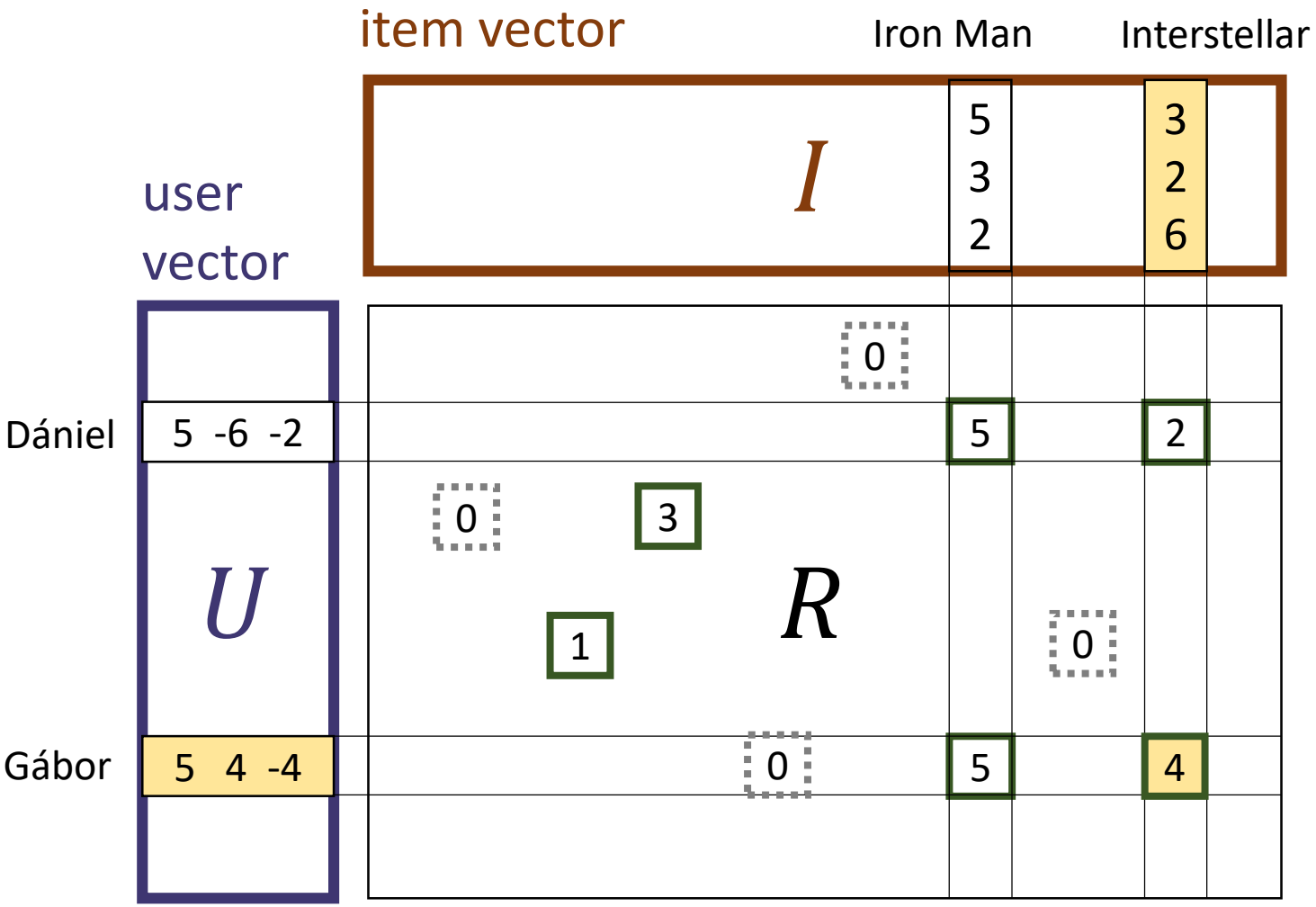
Recommendation with matrix factorization



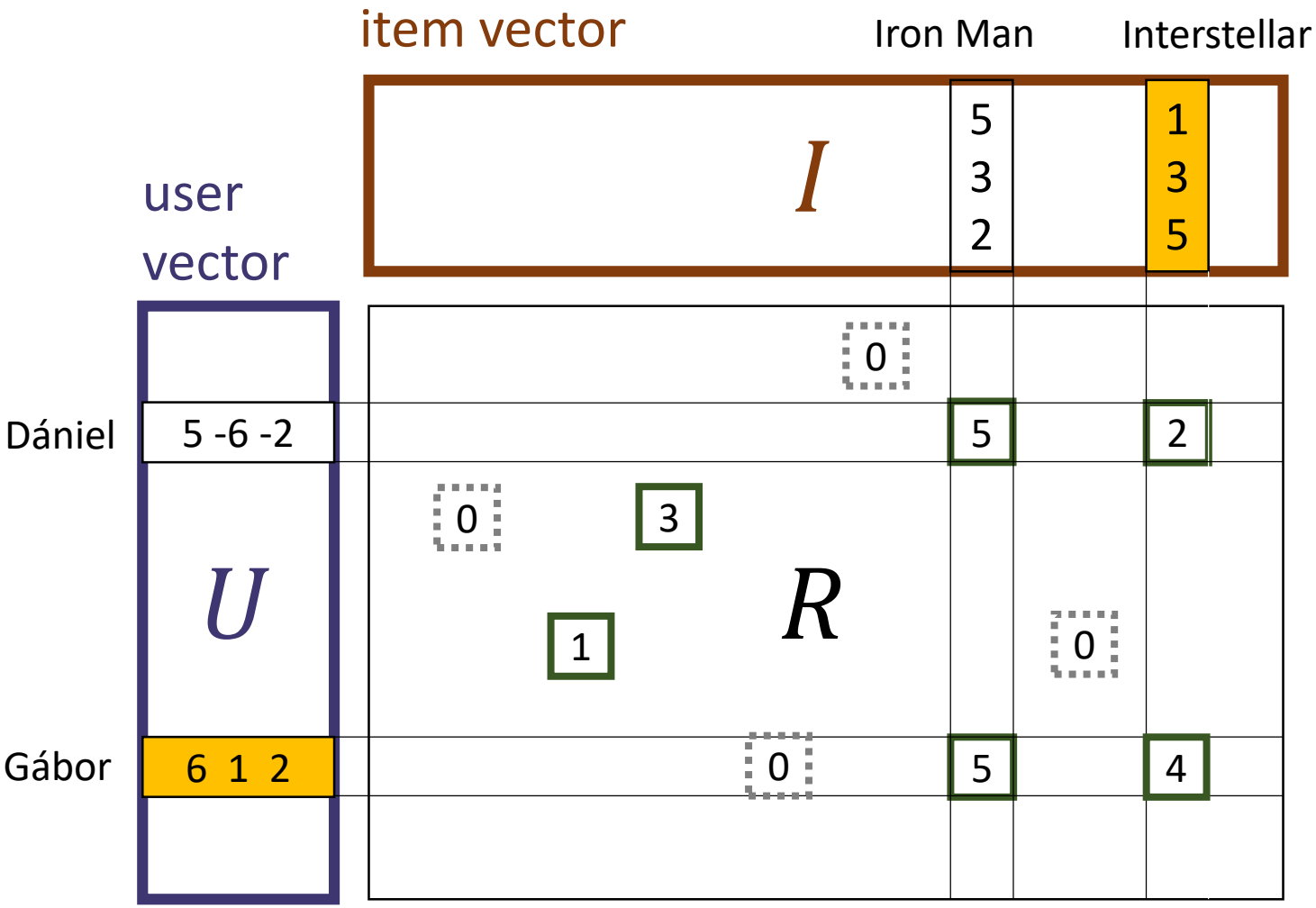
Matrix factorization training



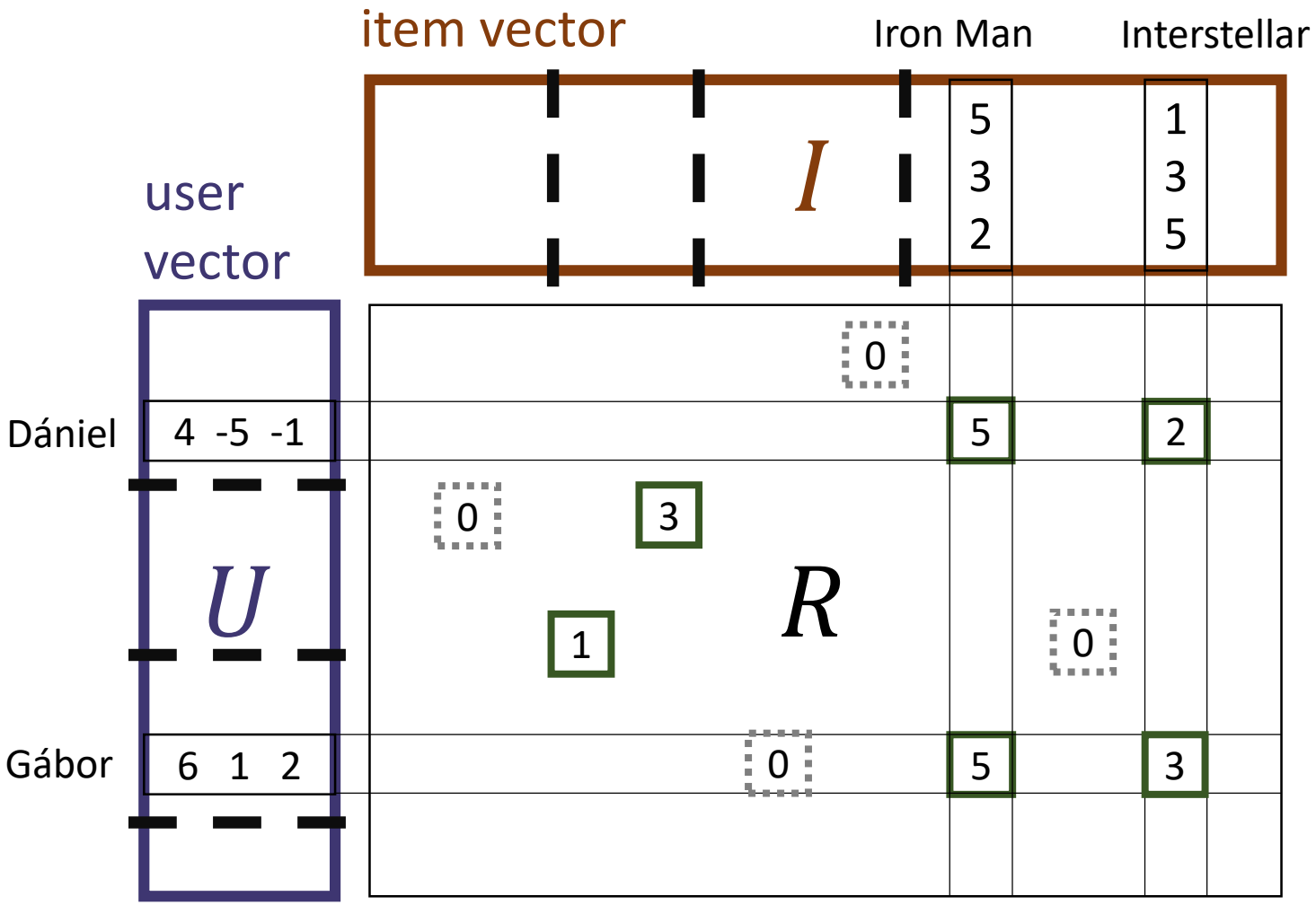
Matrix factorization training



Matrix factorization training



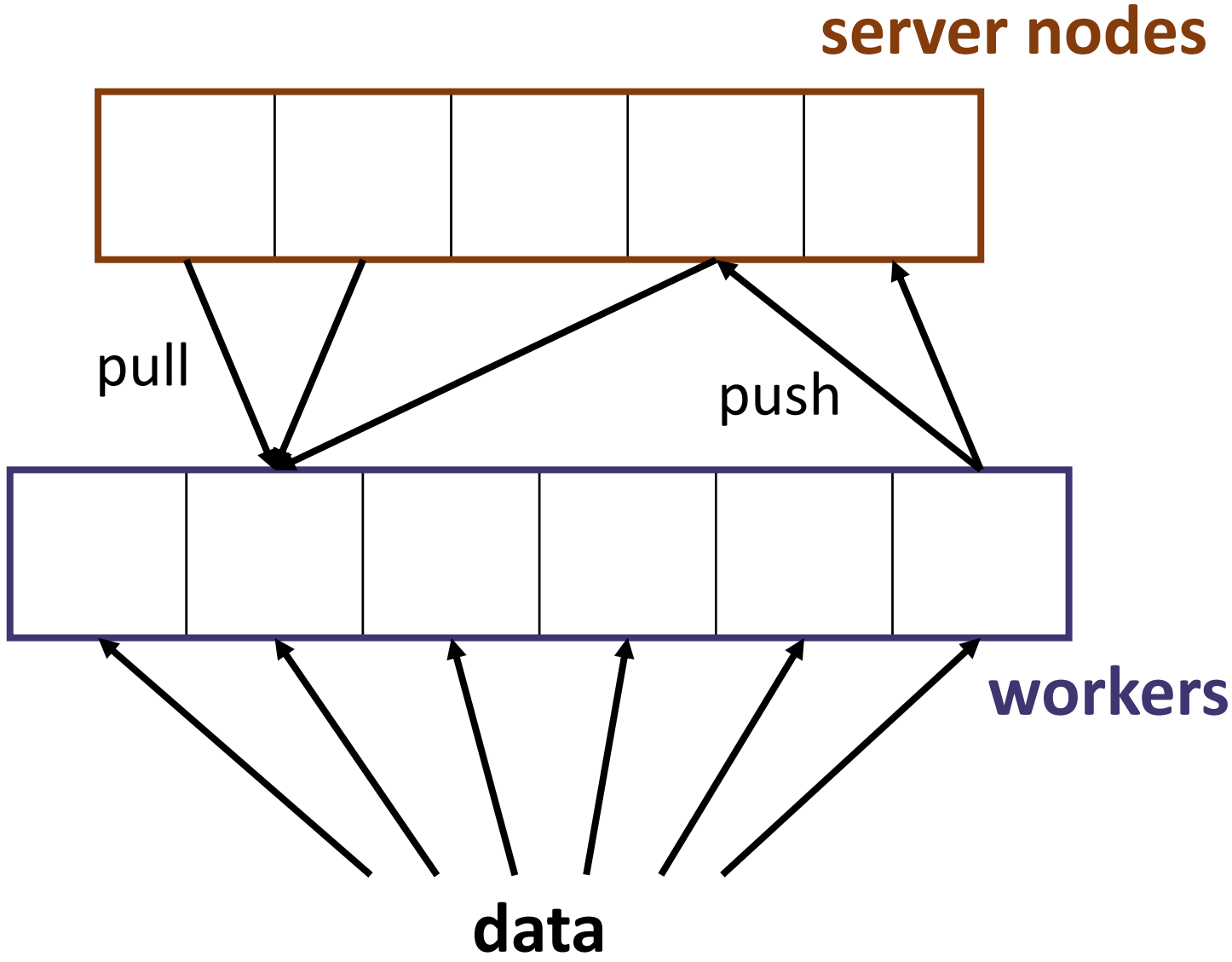
Matrix factorization training



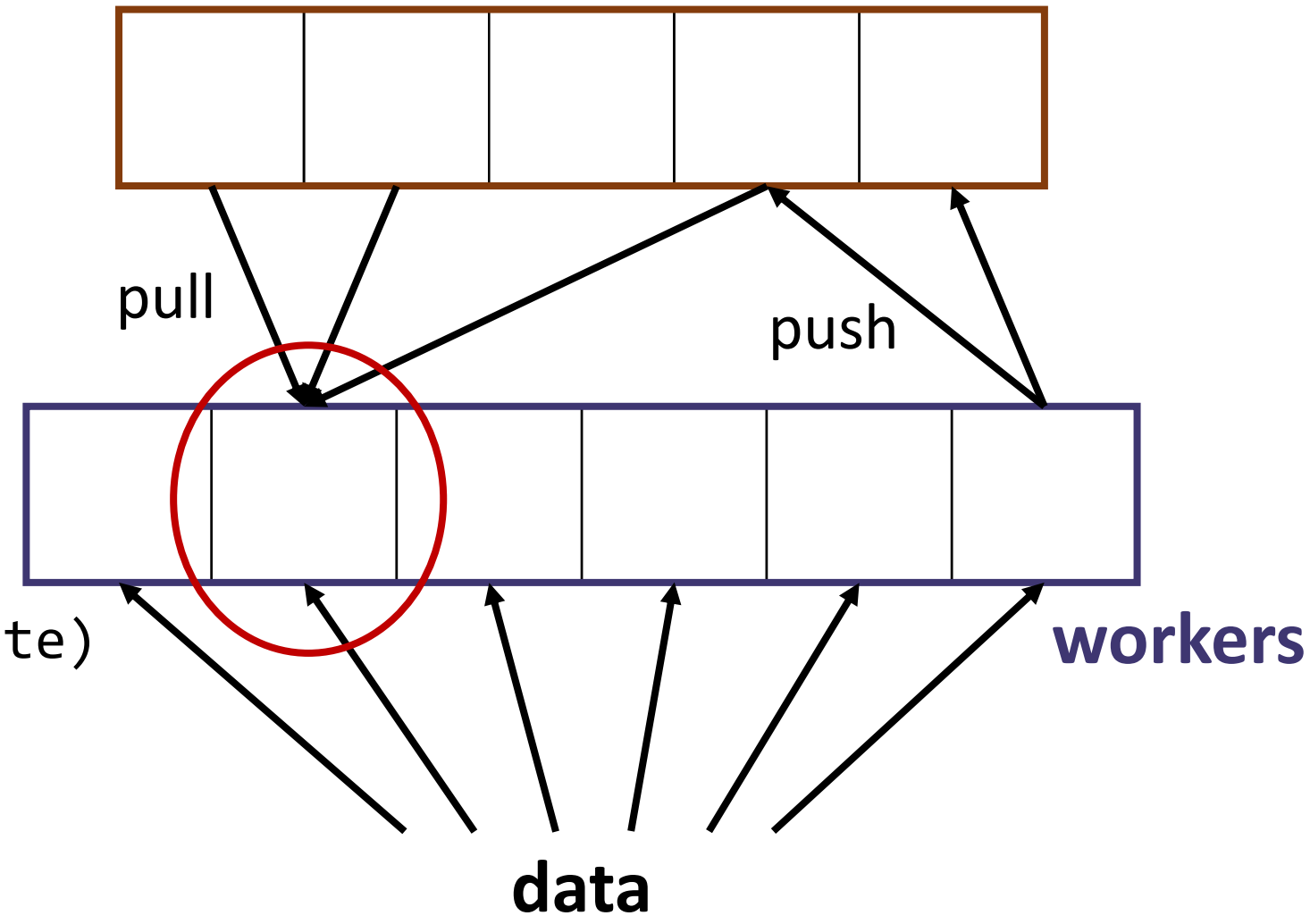
model-parallel

Parameter Server on Flink

Parameter Server

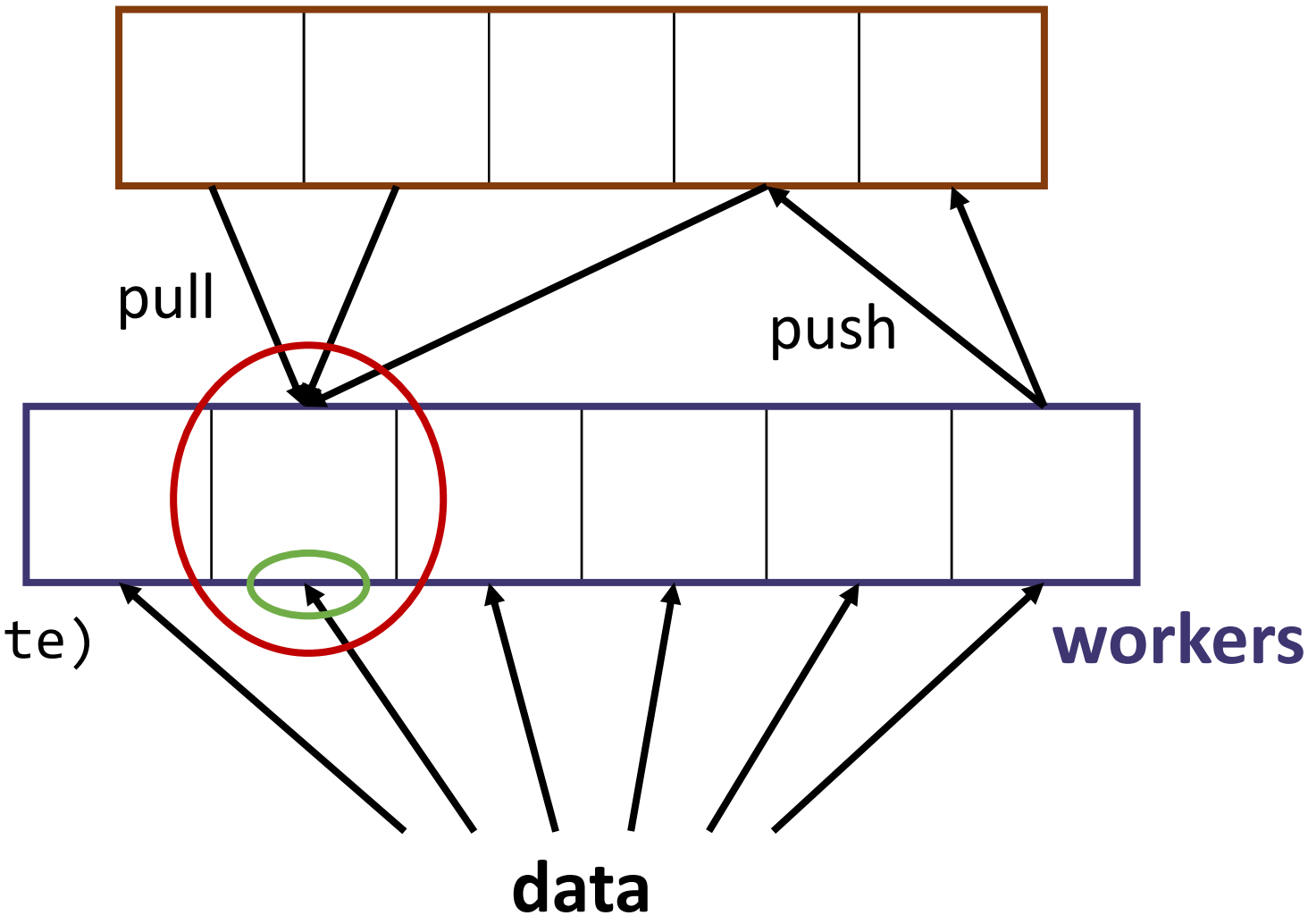


Parameter Server API



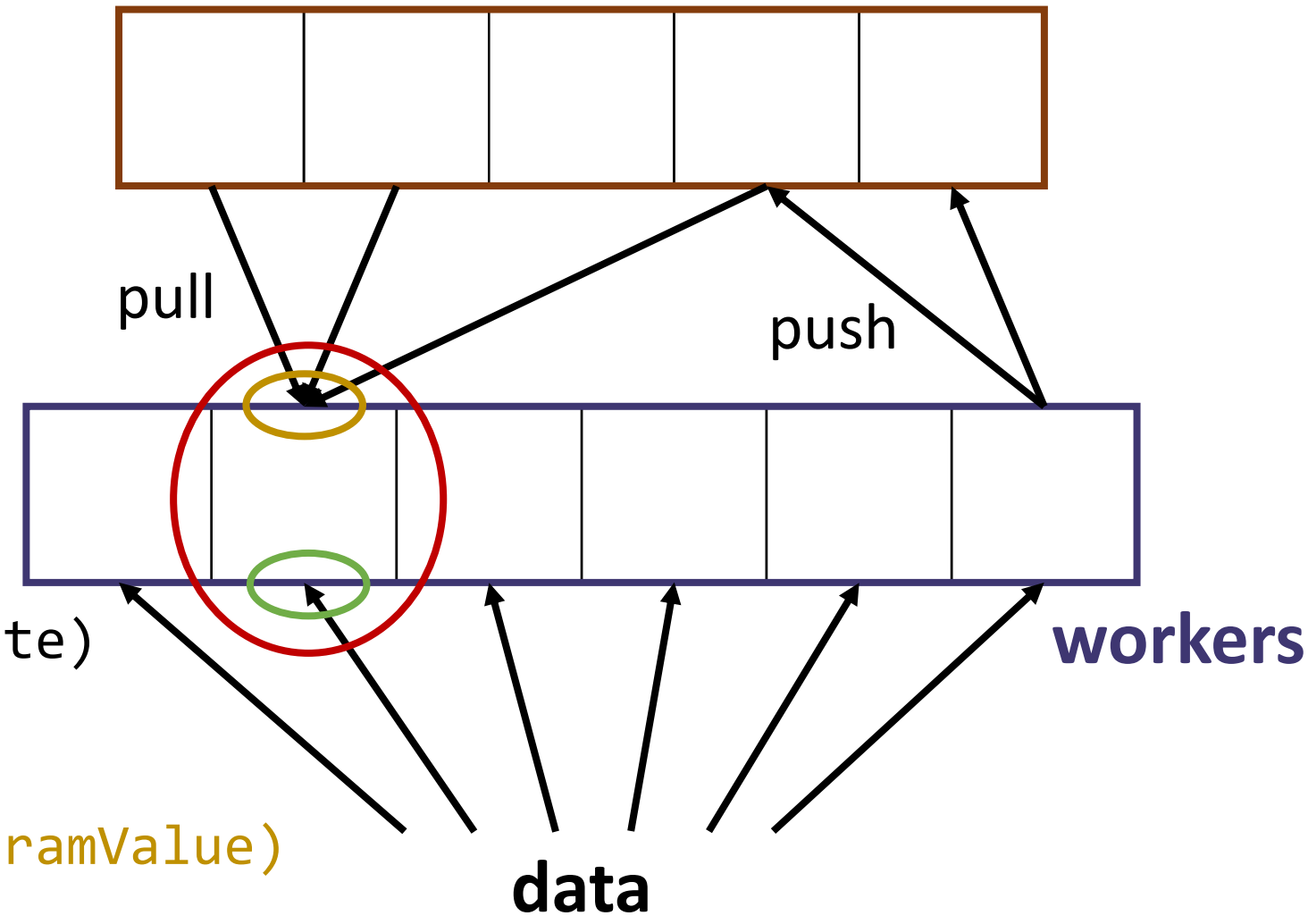

```
ps.pull(paramId)
ps.push(paramId, paramUpdate)
```

Parameter Server API



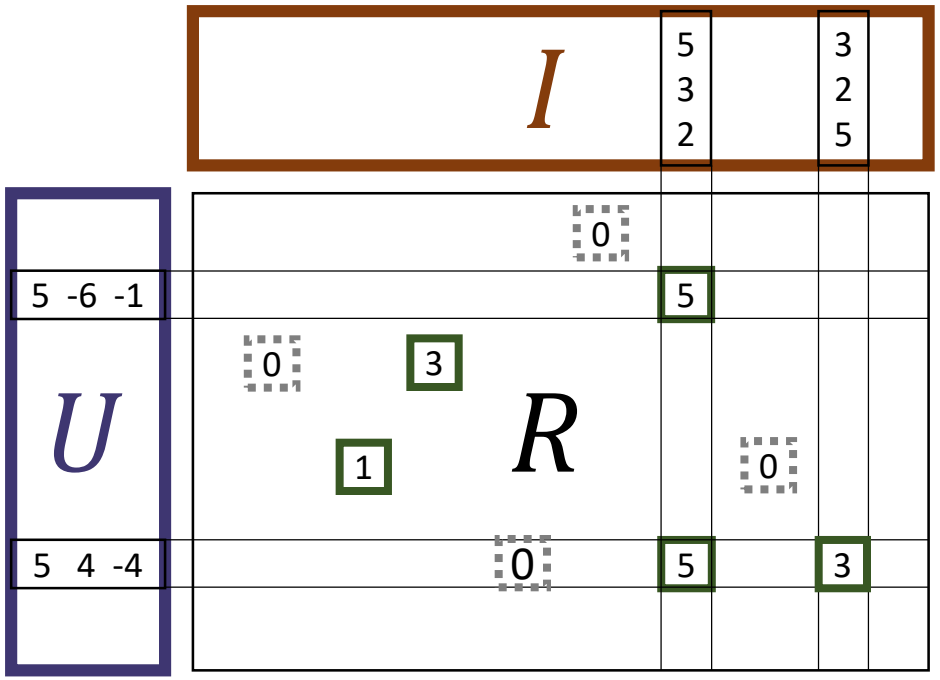

```
ps.pull(paramId)
ps.push(paramId, paramUpdate)

def onRecv(data): Unit
```

Parameter Server API

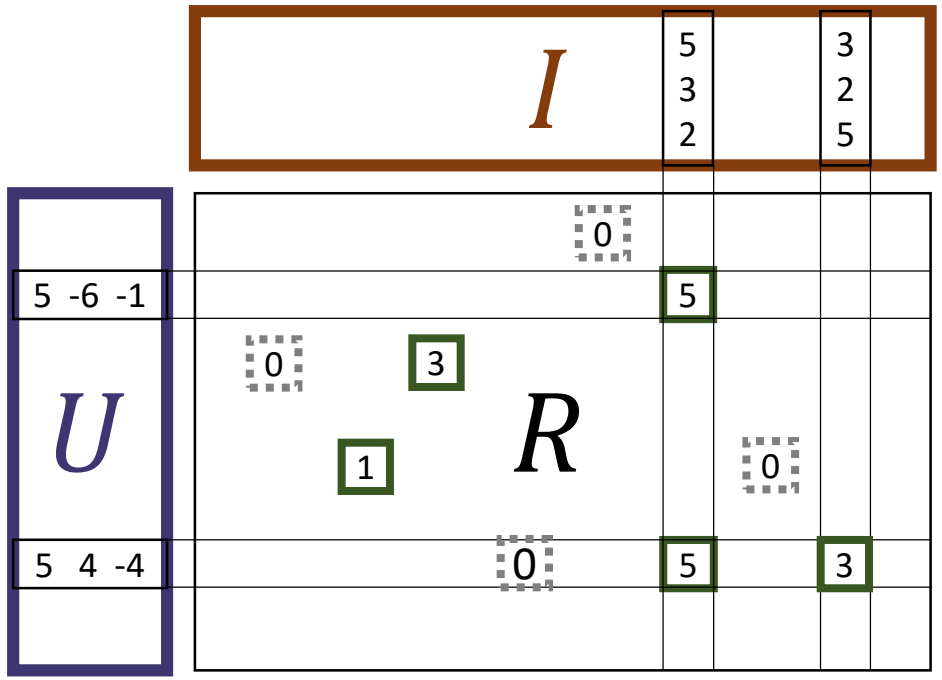


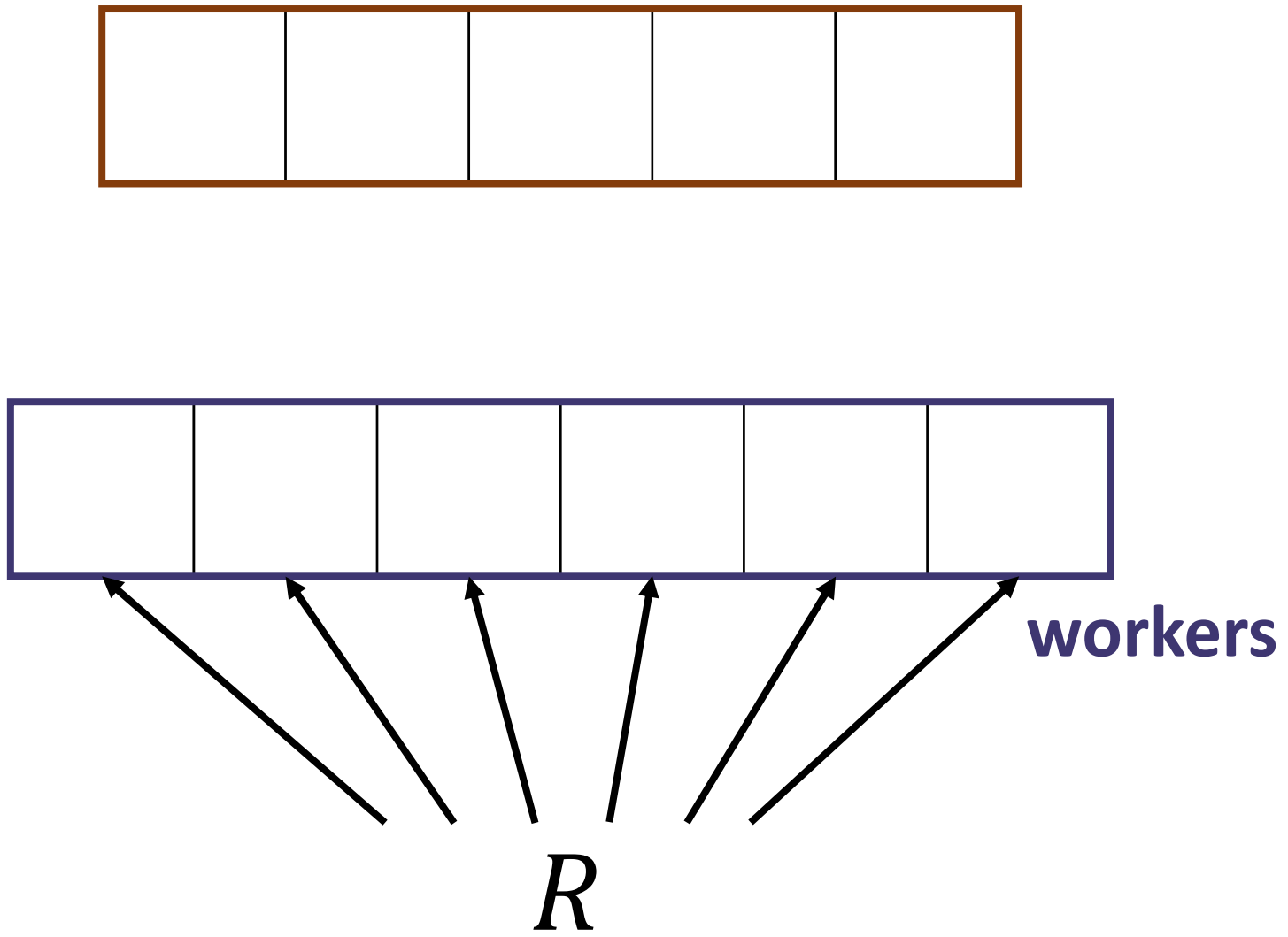

```
ps.pull(paramId)
ps.push(paramId, paramUpdate)

def onRecv(data): Unit
def onPullRecv(paramId, paramValue)
```

Matrix factorization with Parameter Server

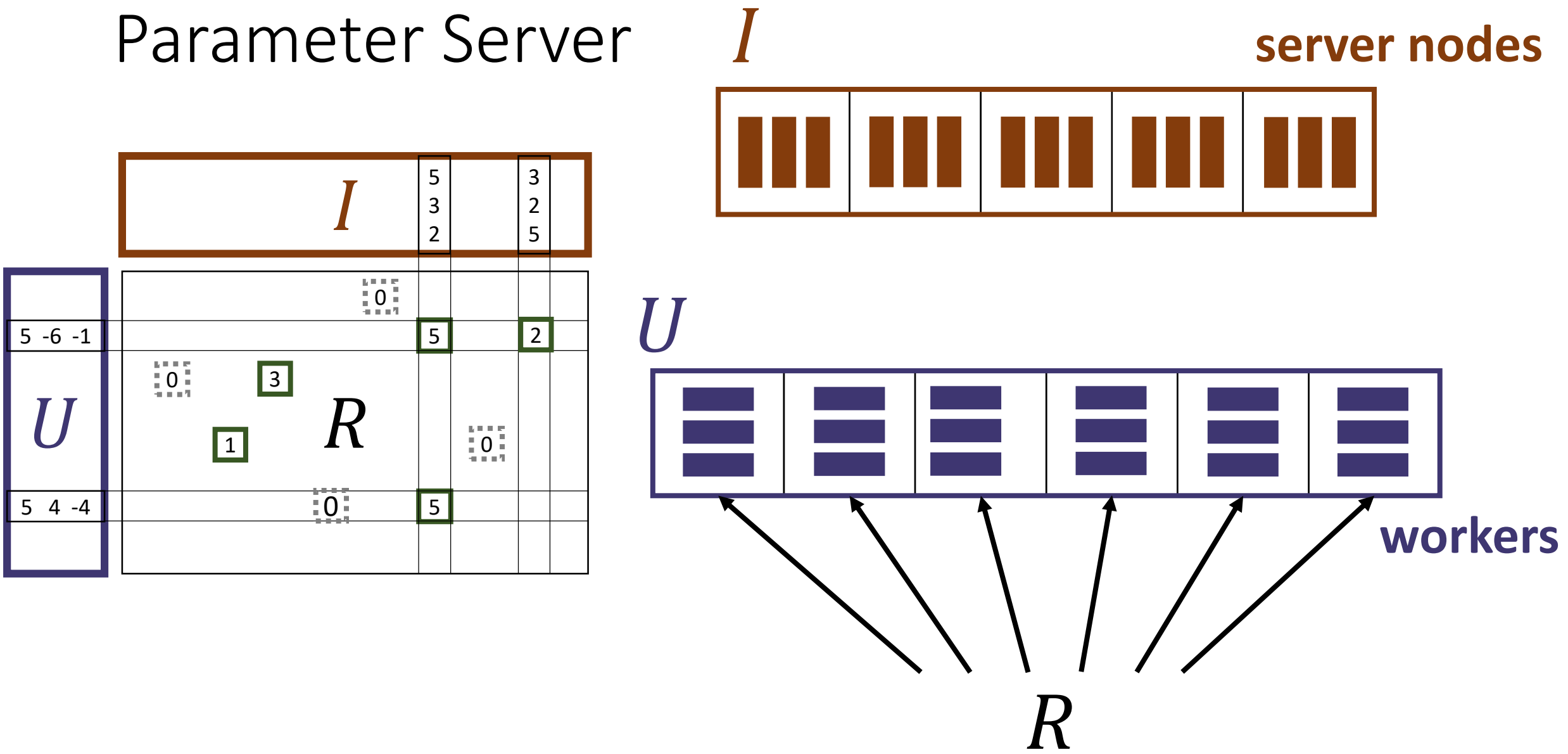


Matrix factorization with Parameter Server

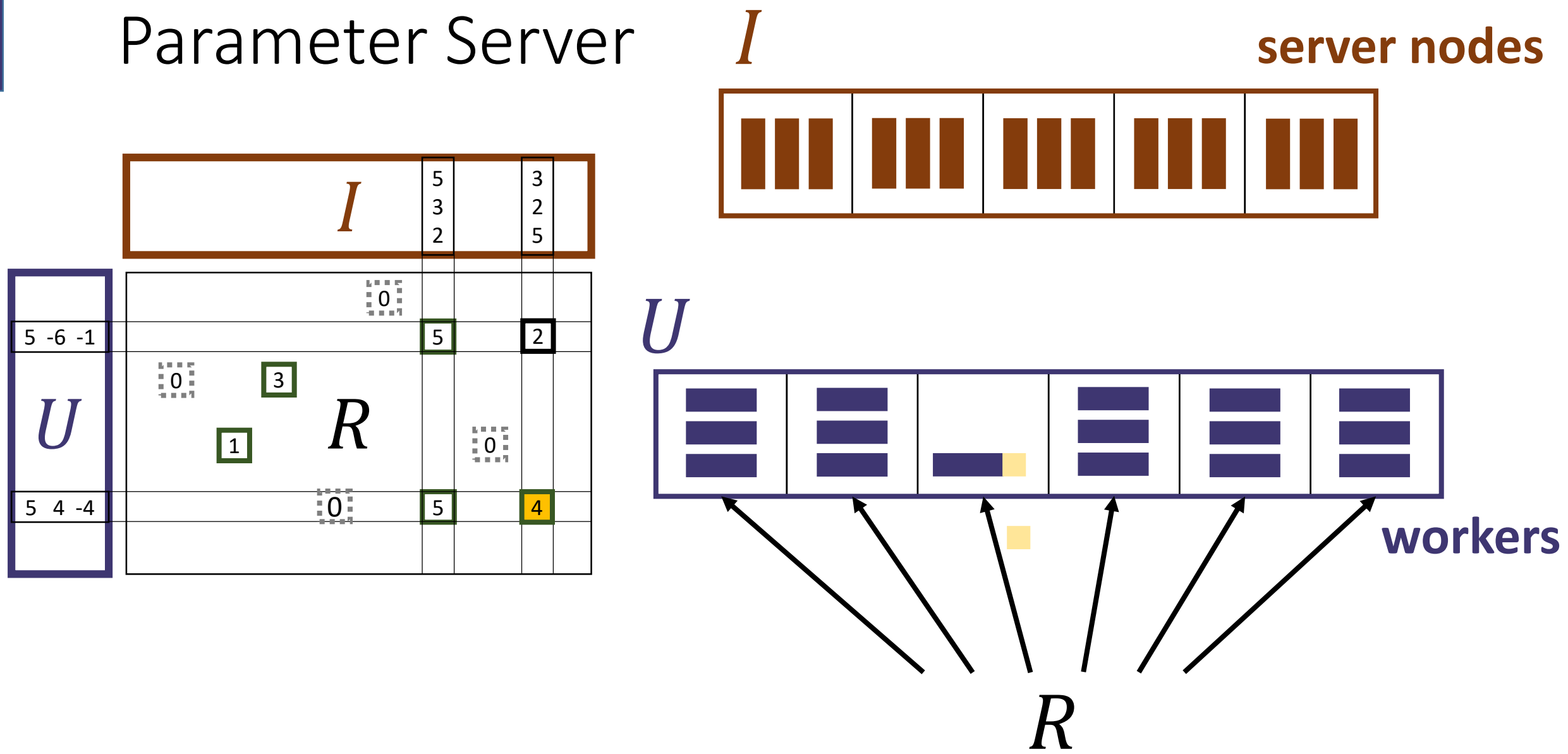




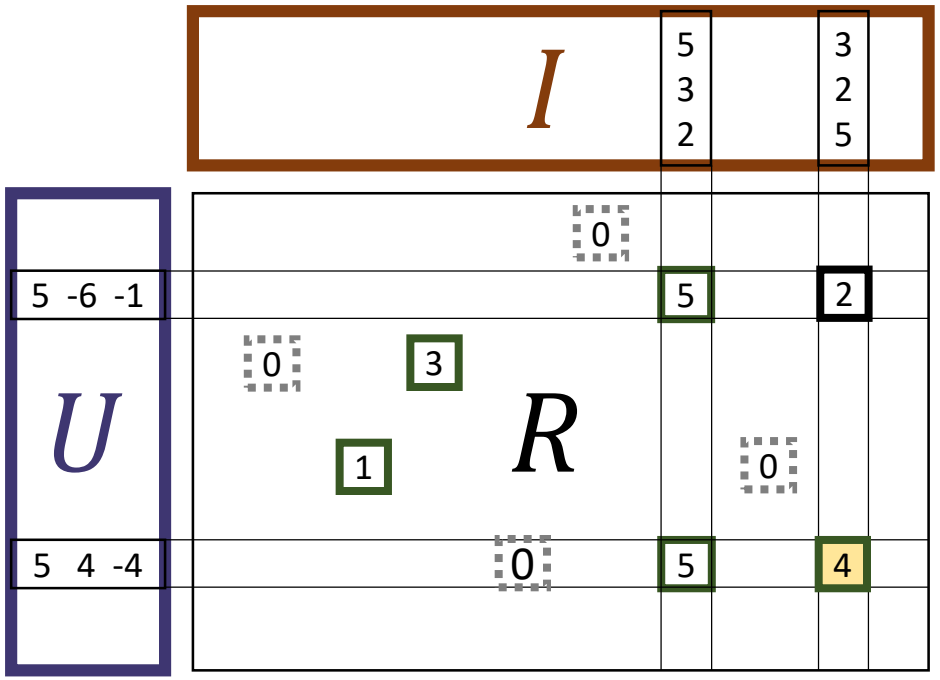
Matrix factorization with Parameter Server

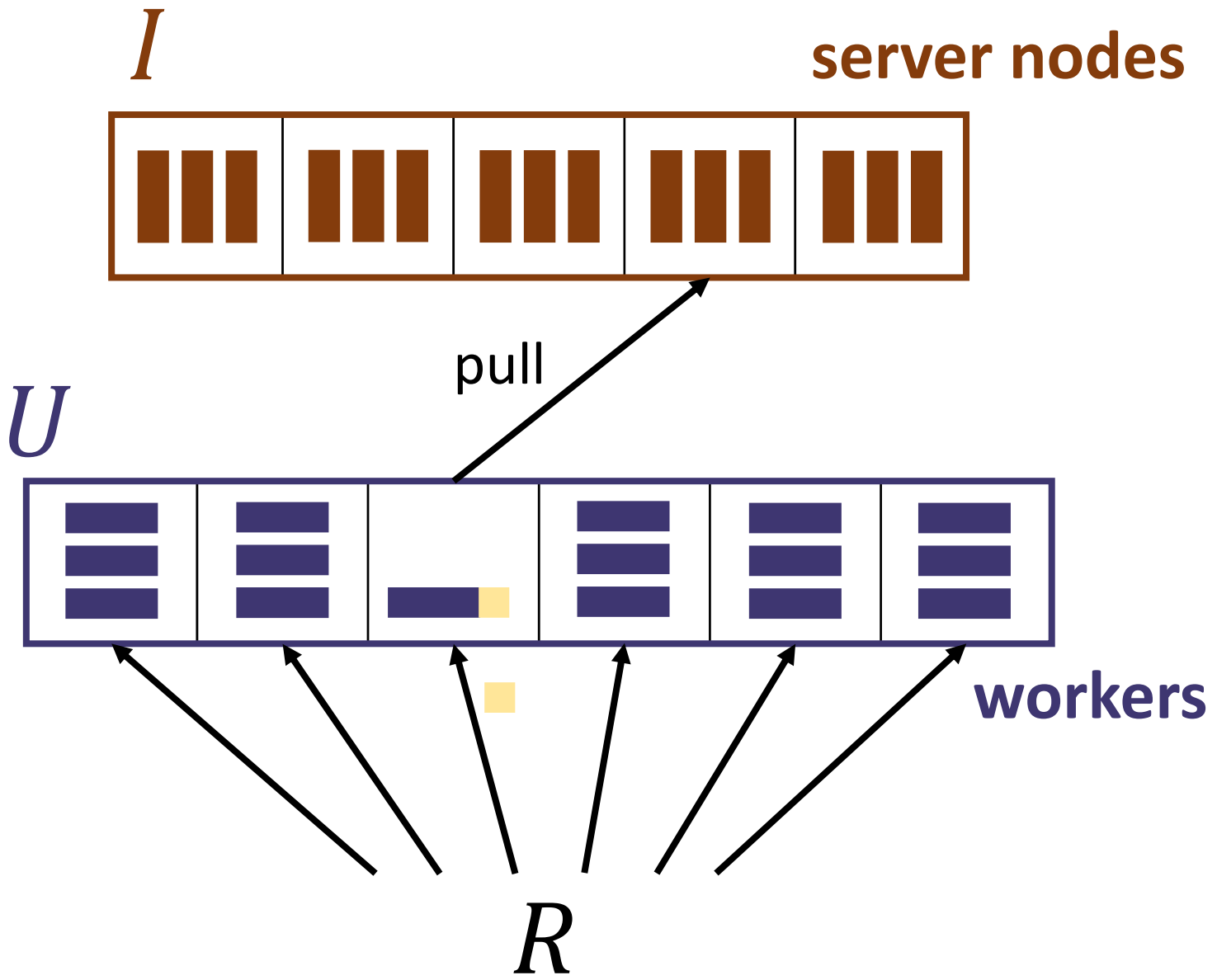


Matrix factorization with Parameter Server

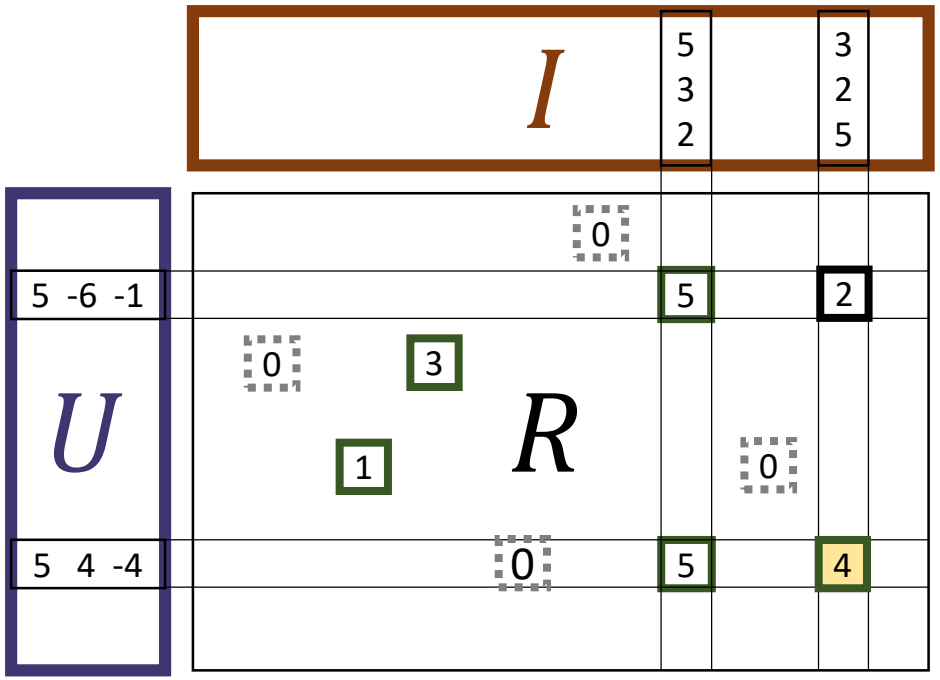


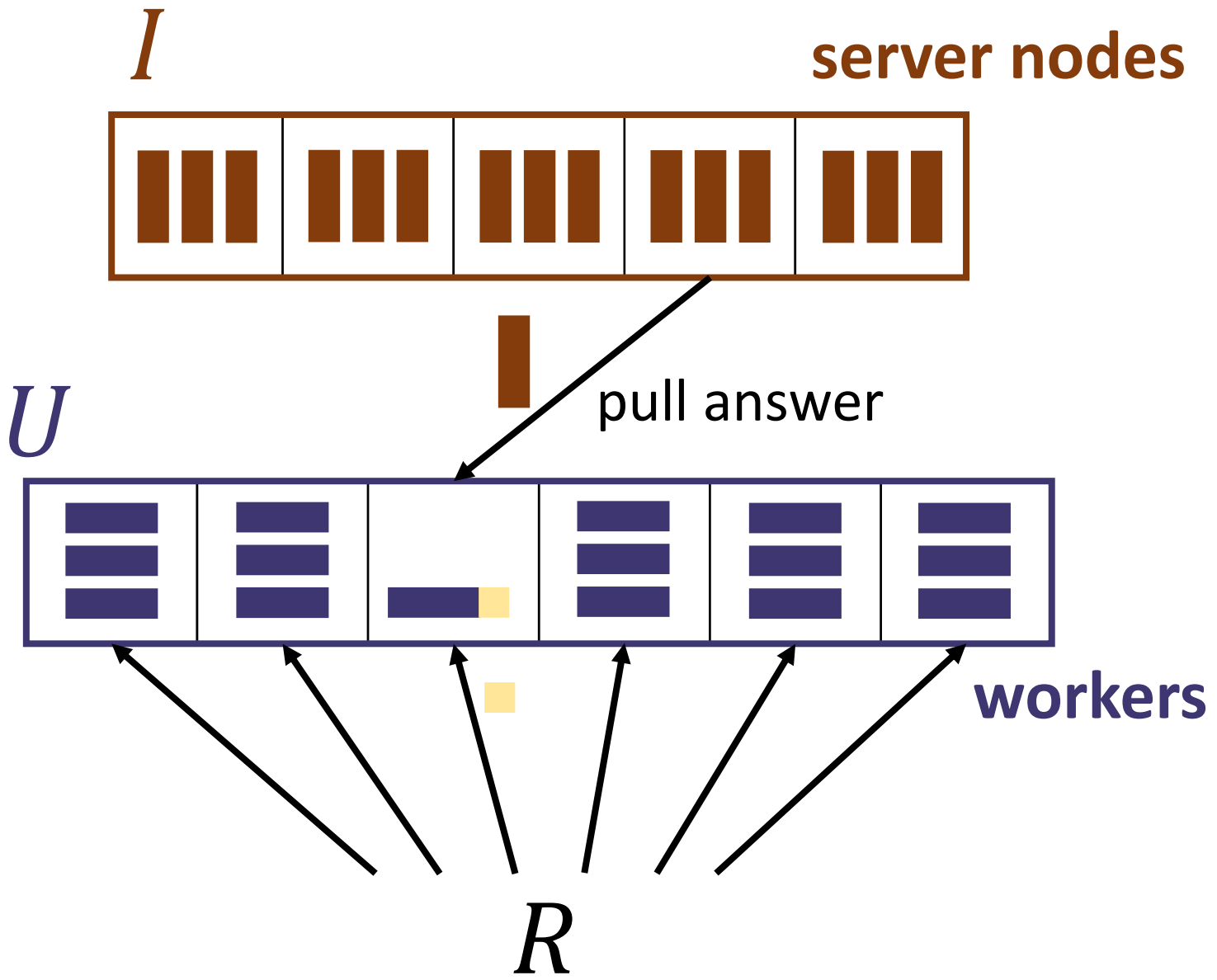
Matrix factorization with Parameter Server



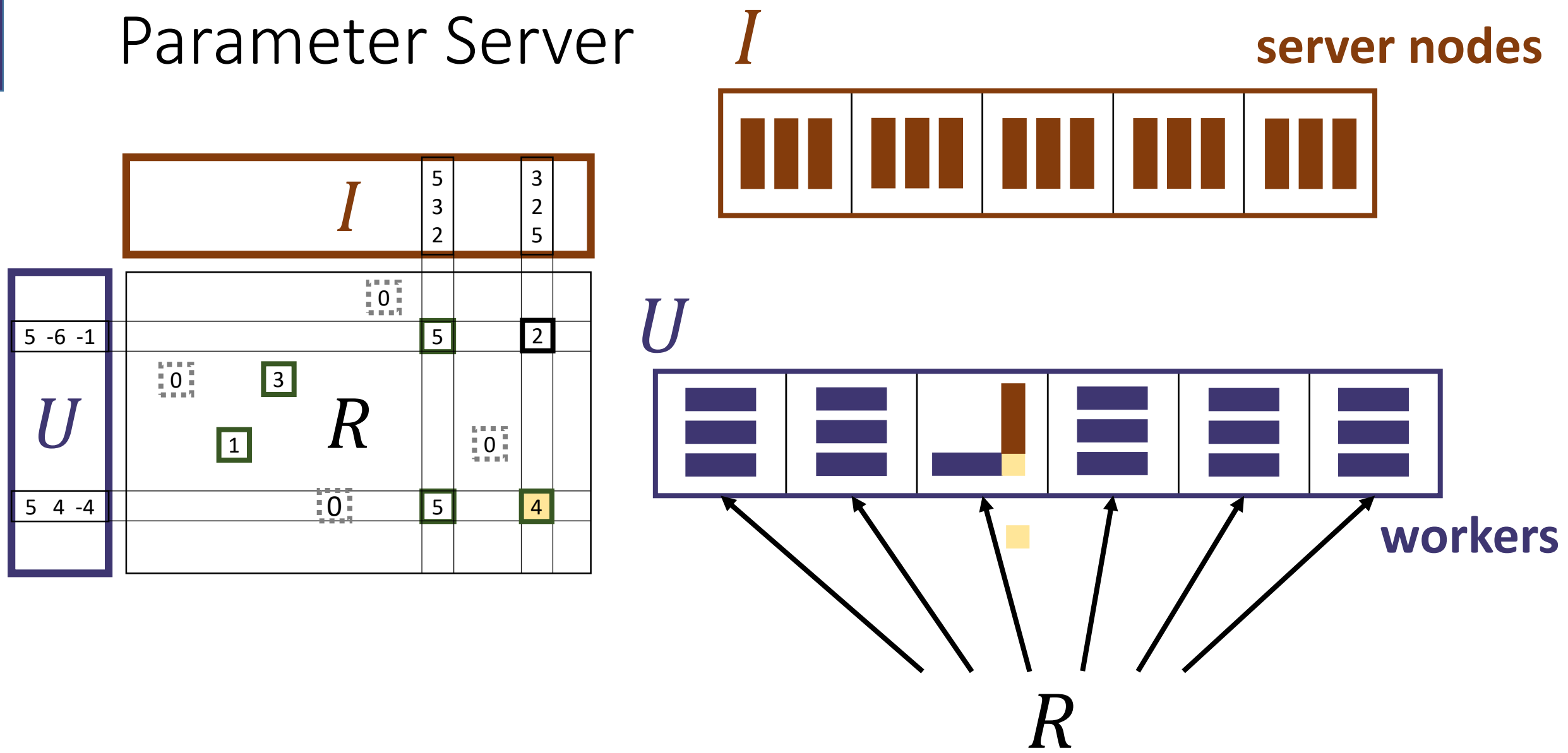


Matrix factorization with Parameter Server

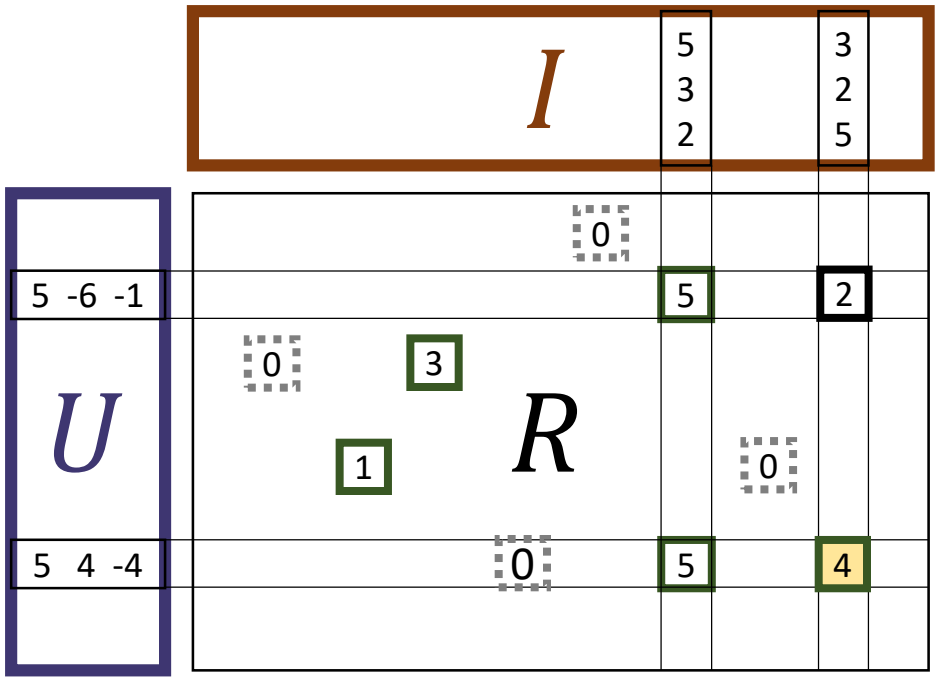


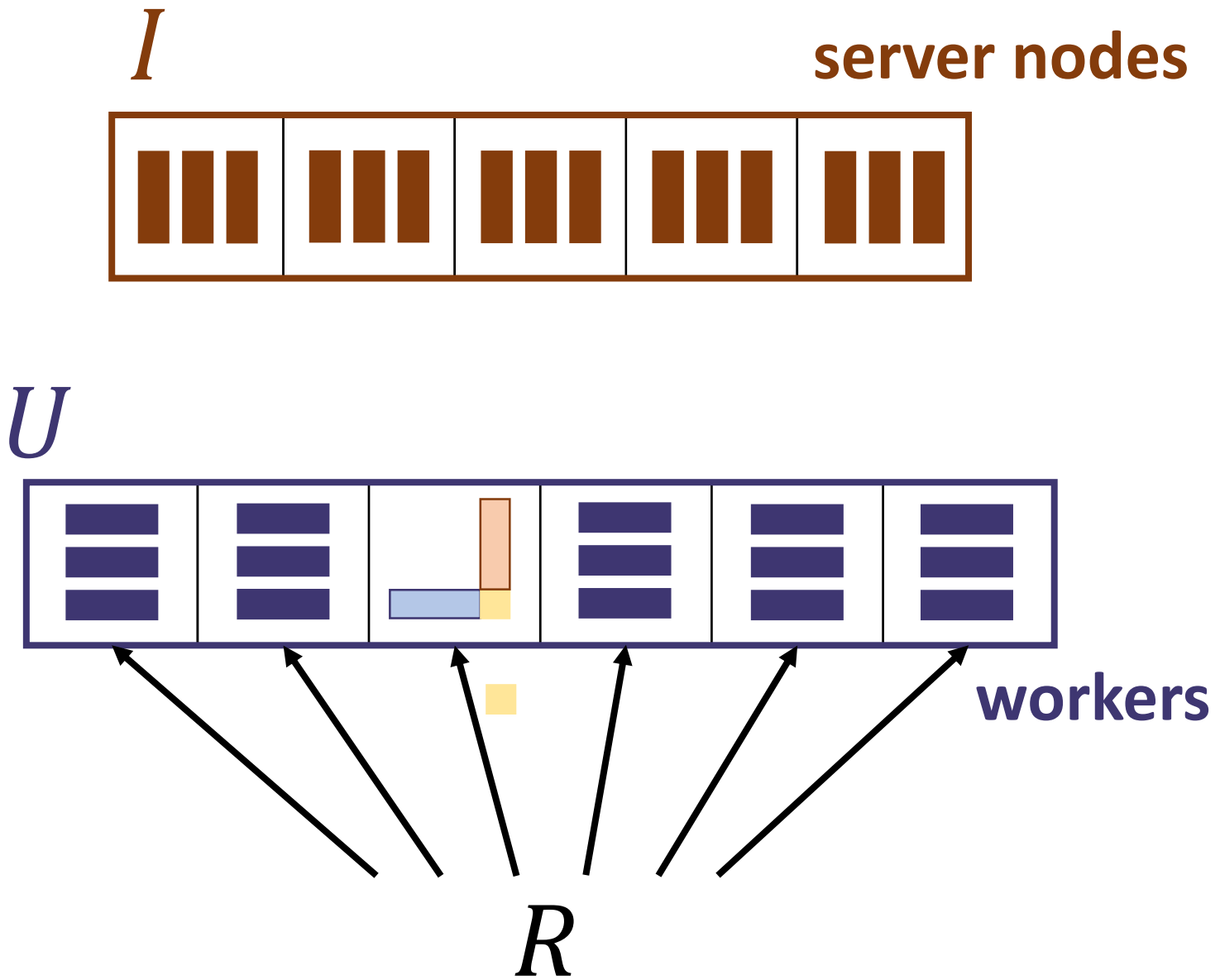


Matrix factorization with Parameter Server

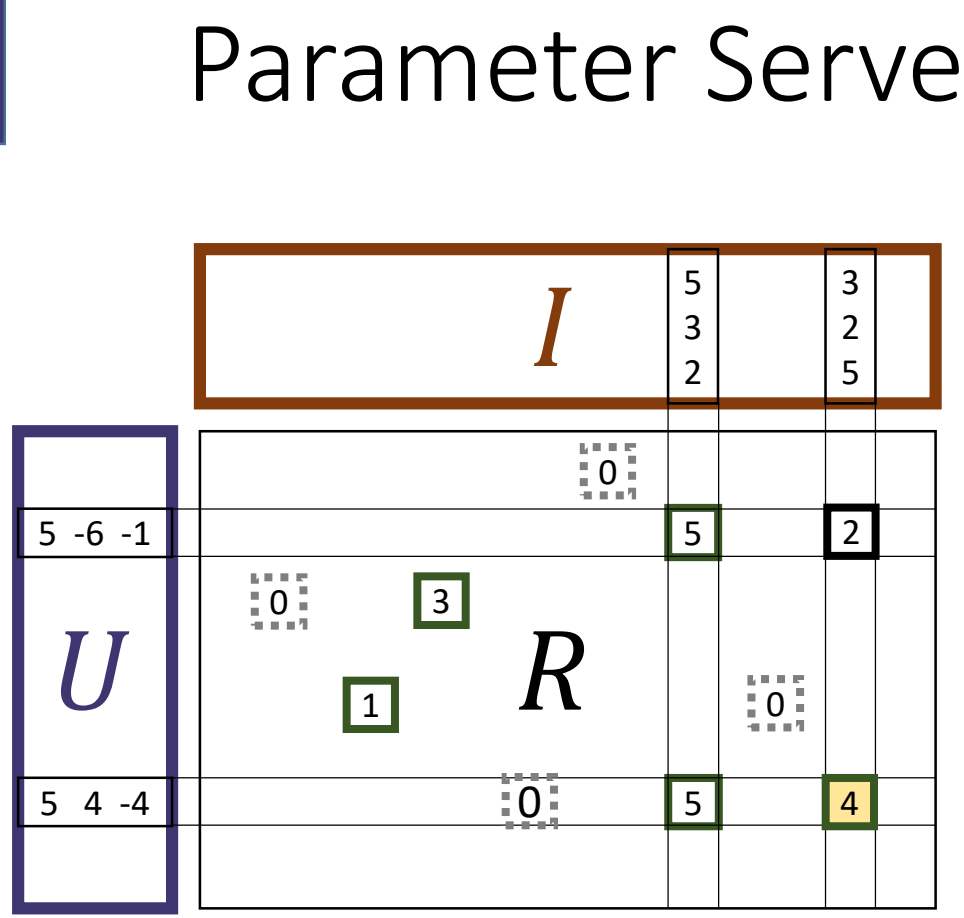


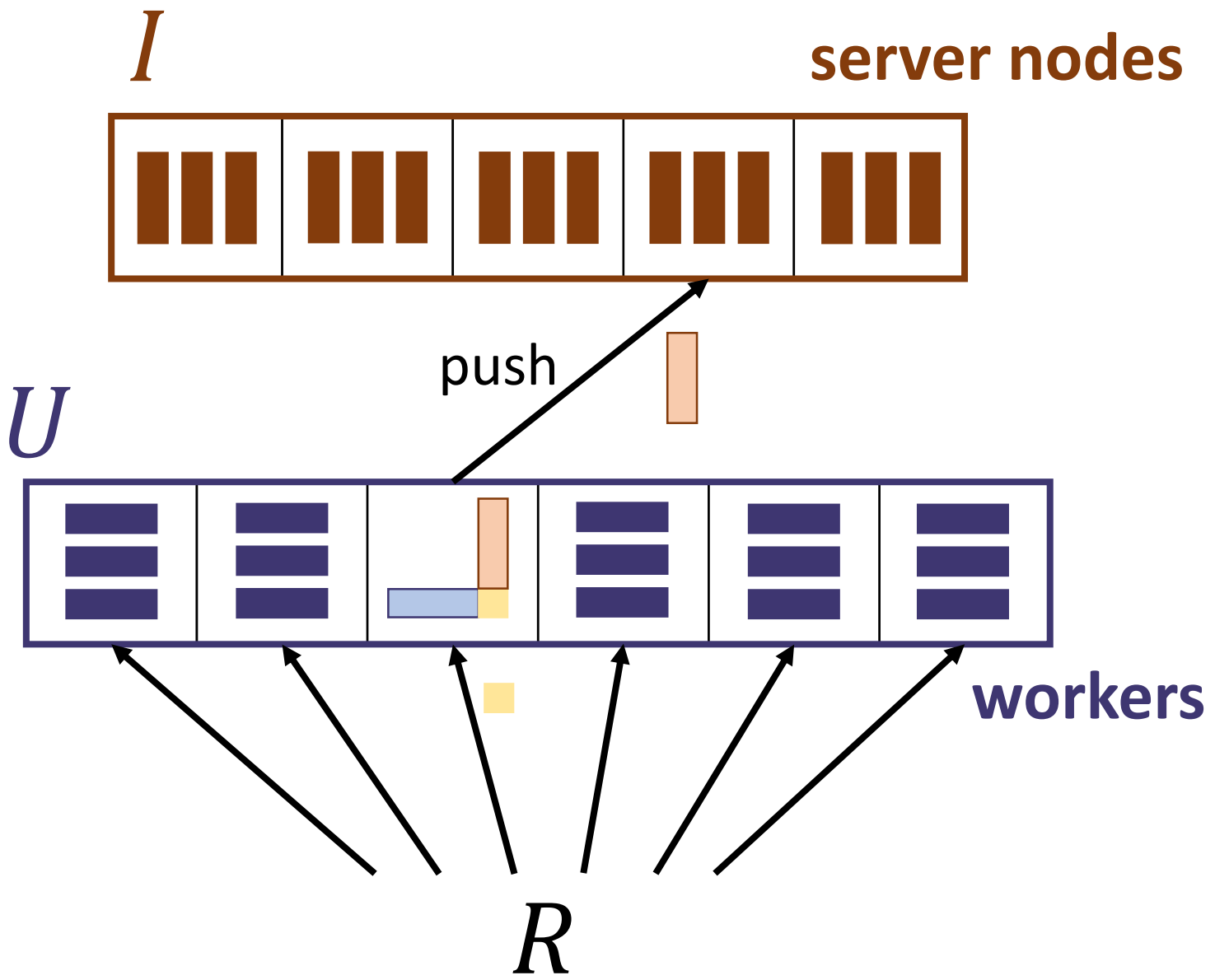
Matrix factorization with Parameter Server



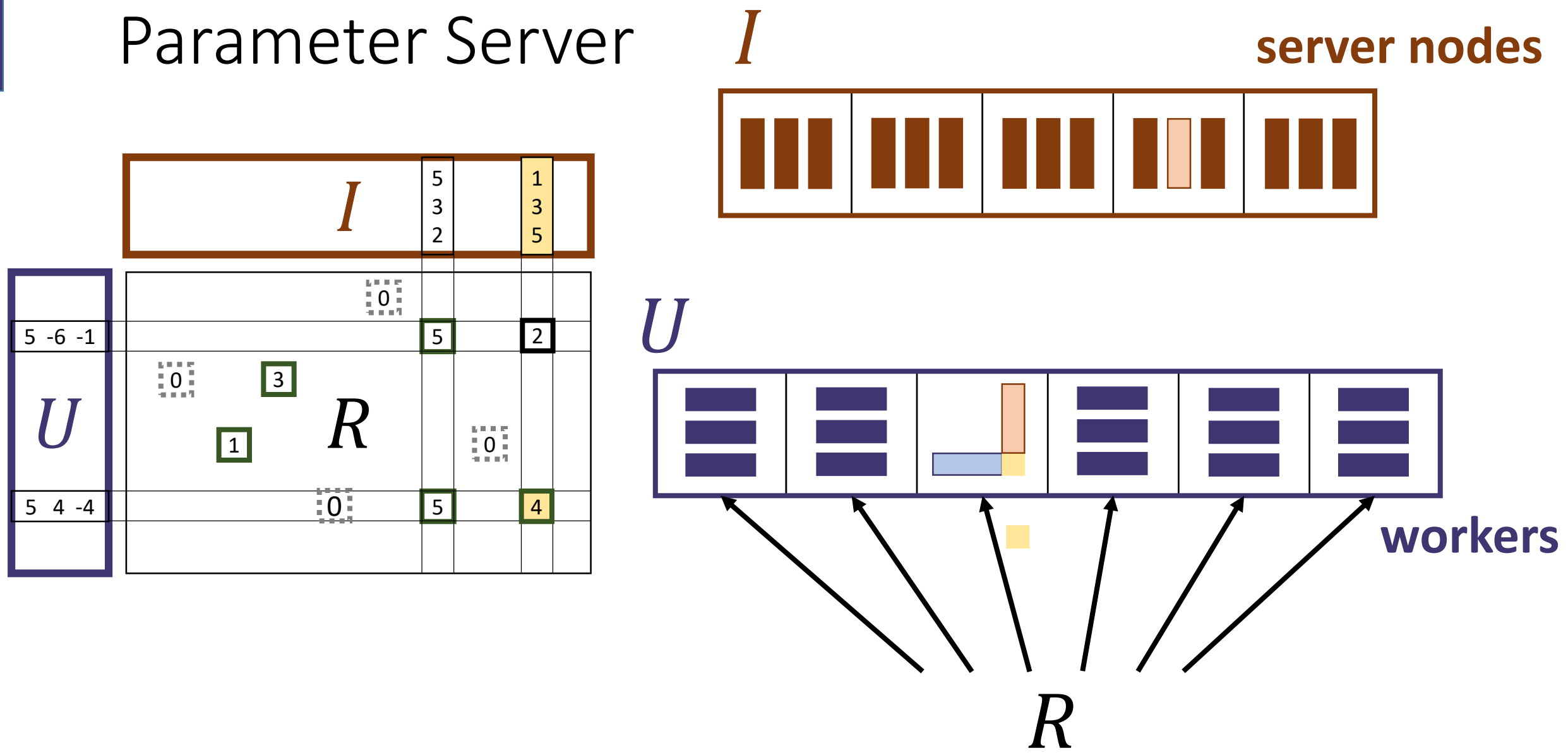


Matrix factorization with Parameter Server

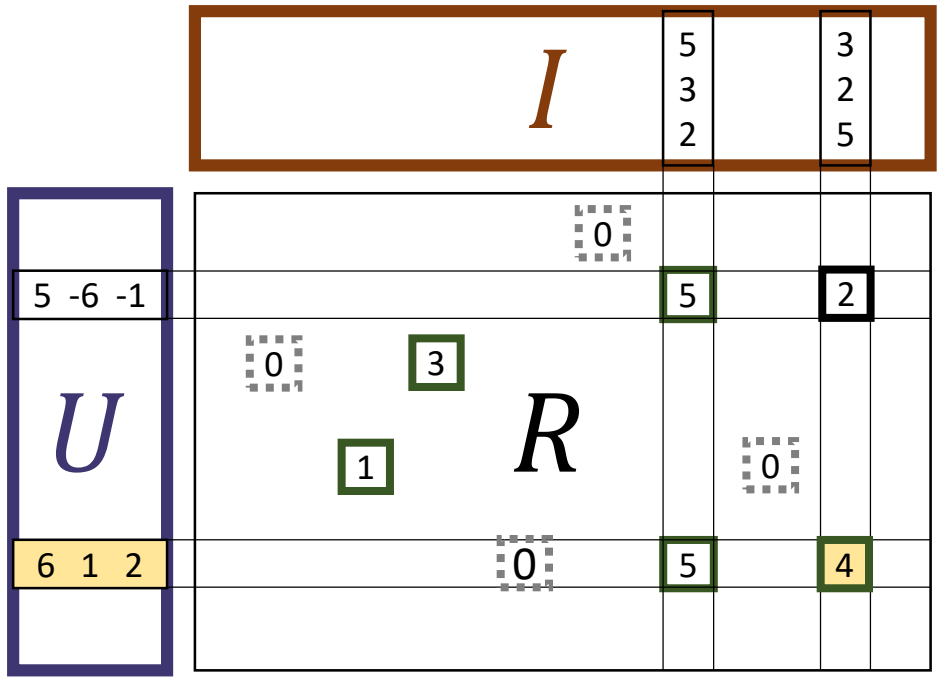


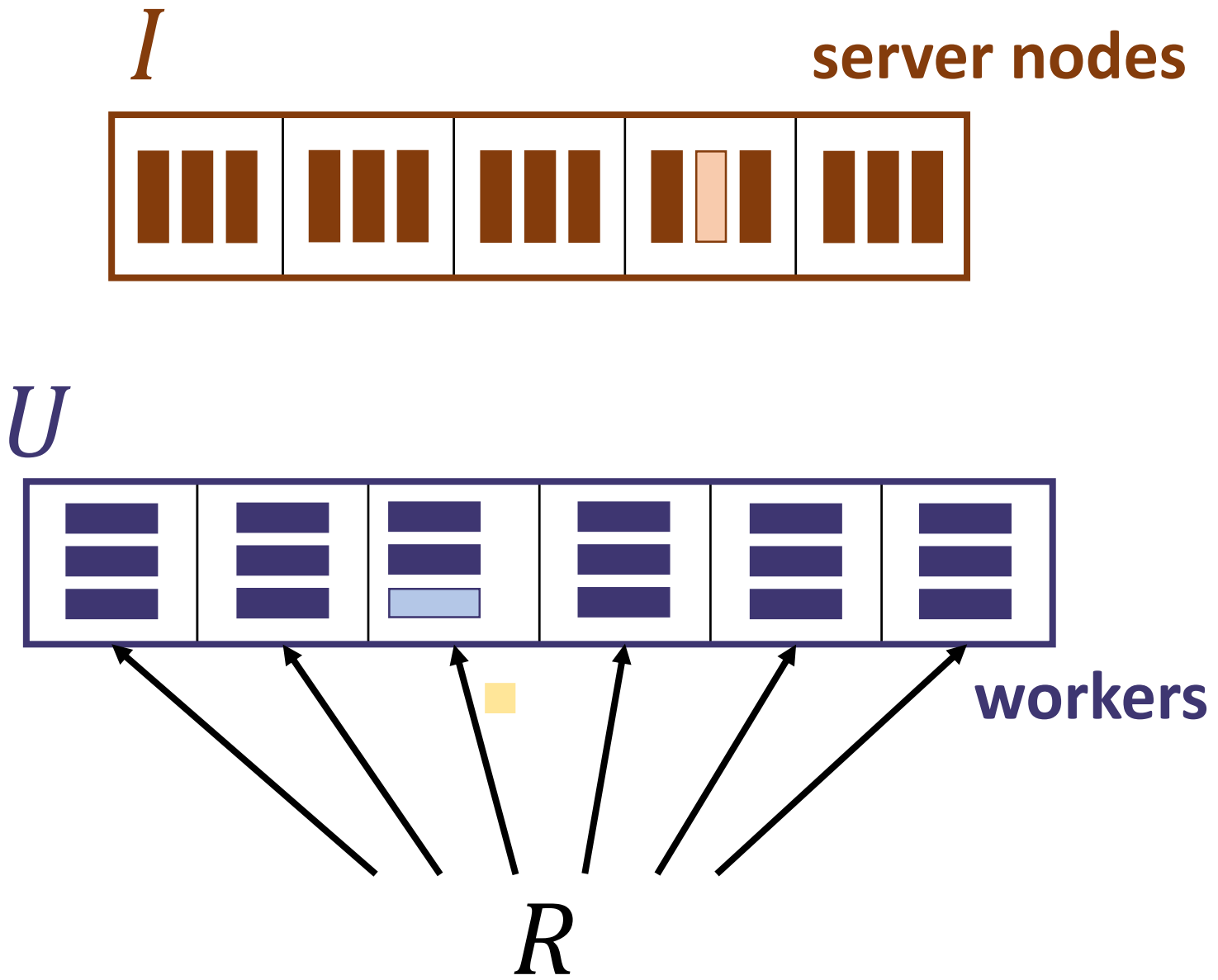


Matrix factorization with Parameter Server



Matrix factorization with Parameter Server





MTA SZTAKI

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
```

MTA SZTAKI

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
```

MTA SZTAKI

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
```

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
  val itemId = paramId
  val item = param
```

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
  val itemId = paramId
  val item = param

  val (r, userId, _) =
    waitQueues(itemId).pop()
```

MTA SZTAKI

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
  val itemId = paramId
  val item = param

  val (r, userId, _) =
    waitQueues(itemId).pop()
  val user = users(userId)
```

MTA SZTAKI

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
  val itemId = paramId
  val item = param

  val (r, userId, _) =
    waitQueues(itemId).pop()
  val user = users(userId)
  val (userDelta, itemDelta) =
    updateWithSGD(user, item, r)
```

Matrix Factorization code (SGD)

```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

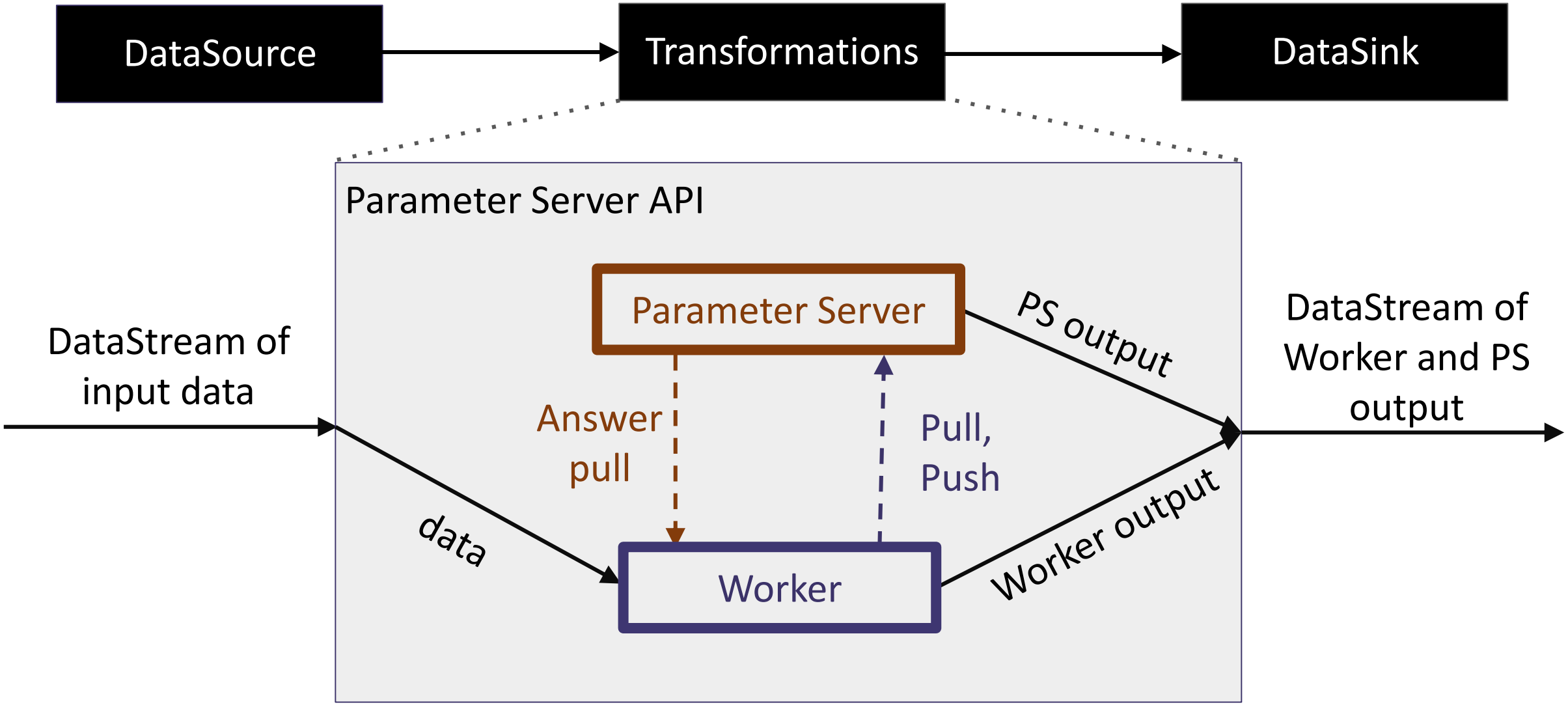
```
def onPullRecv(paramId: Int,
               param: Vector) = {
  val itemId = paramId
  val item = param

  val (r, userId, _) =
    waitQueues(itemId).pop()
  val user = users(userId)
  val (userDelta, itemDelta) =
    updateWithSGD(user, item, r)
  users(userId) += userDelta
```

Matrix Factorization code (SGD)

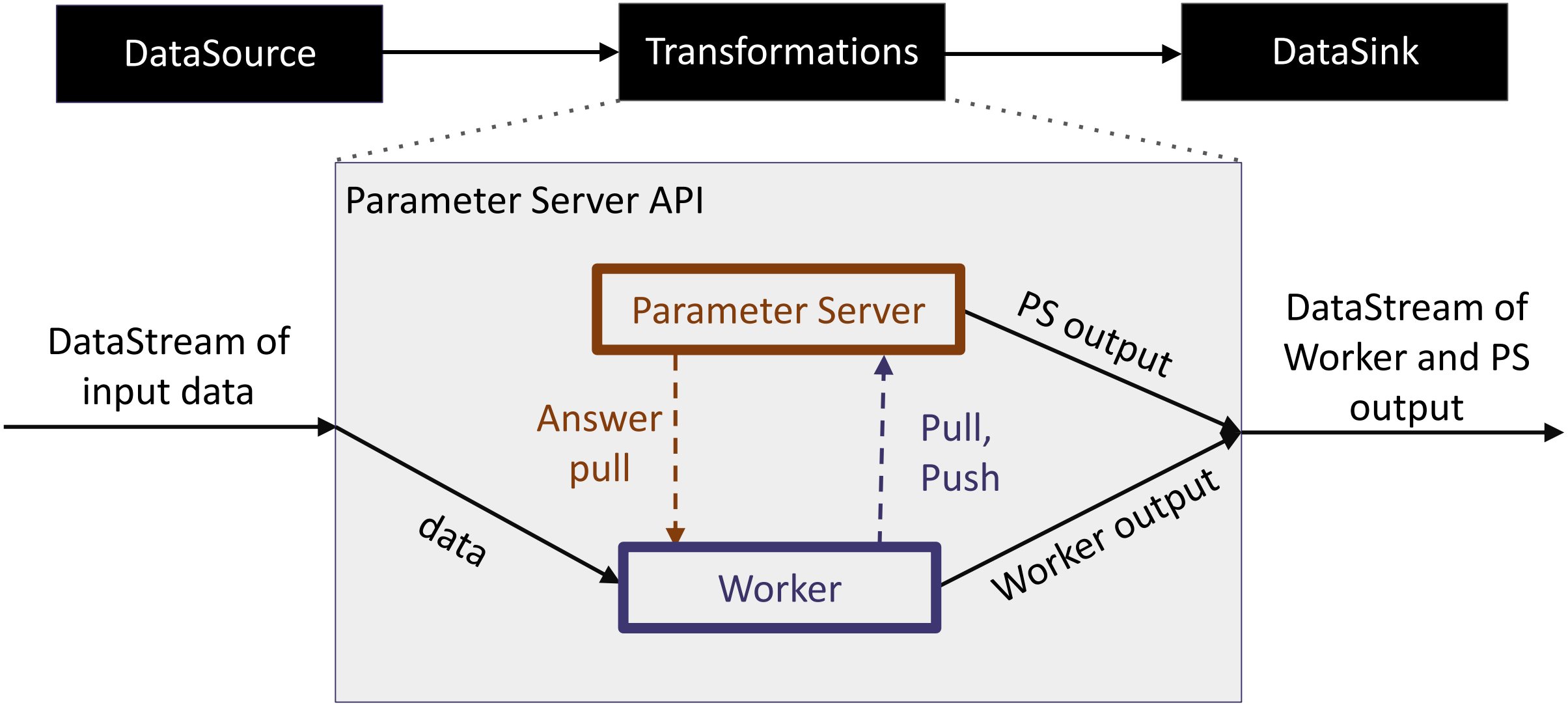
```
def onRecv(r: Rating) = {
  waitQueues(r.itemId).add(r)
  ps.pull(r.itemId)
}
```

```
def onPullRecv(paramId: Int,
               param: Vector) = {
  val itemId = paramId
  val item = param

  val (r, userId, _) =
    waitQueues(itemId).pop()
  val user = users(userId)
  val (userDelta, itemDelta) =
    updateWithSGD(user, item, r)
  users(userId) += userDelta
  ps.push(itemId, itemDelta)
}
```

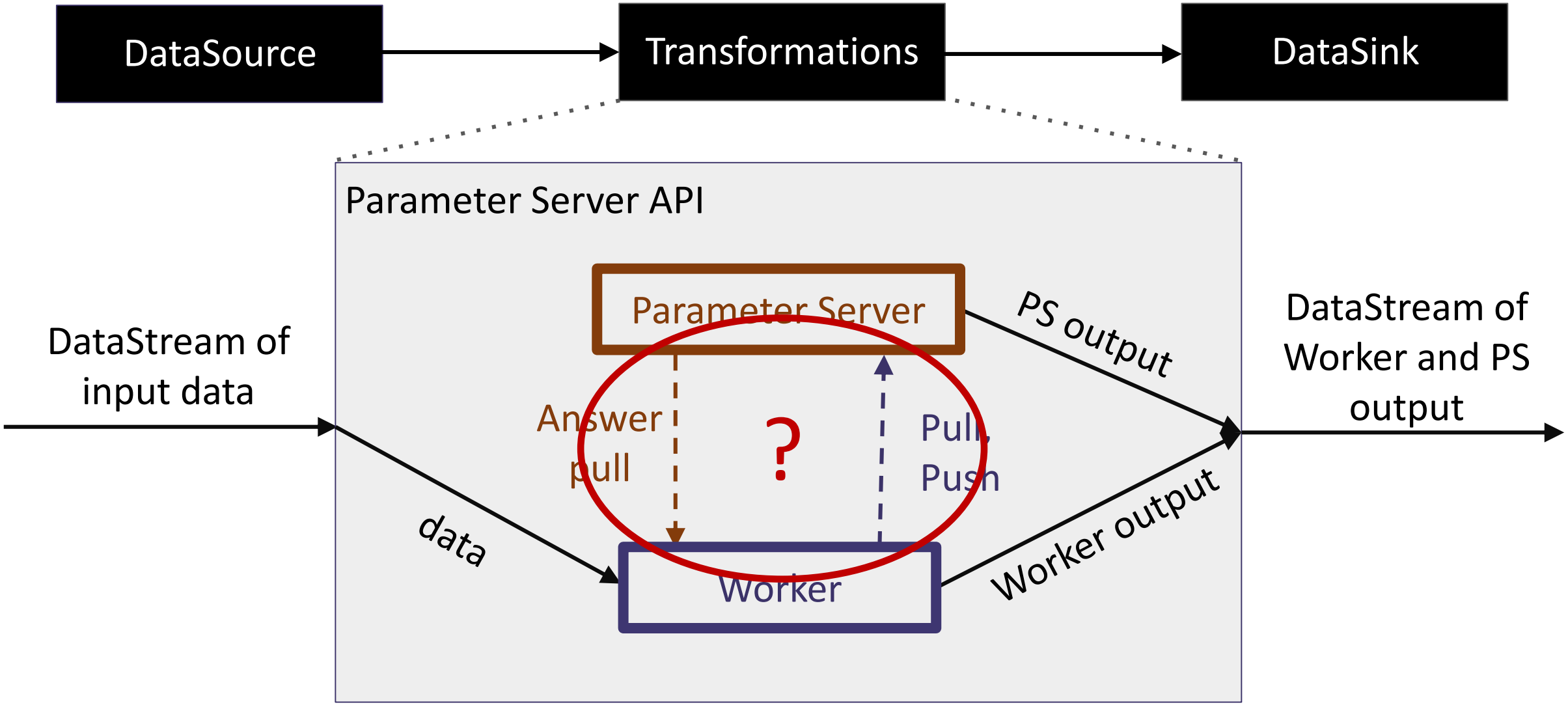
Integration with Flink



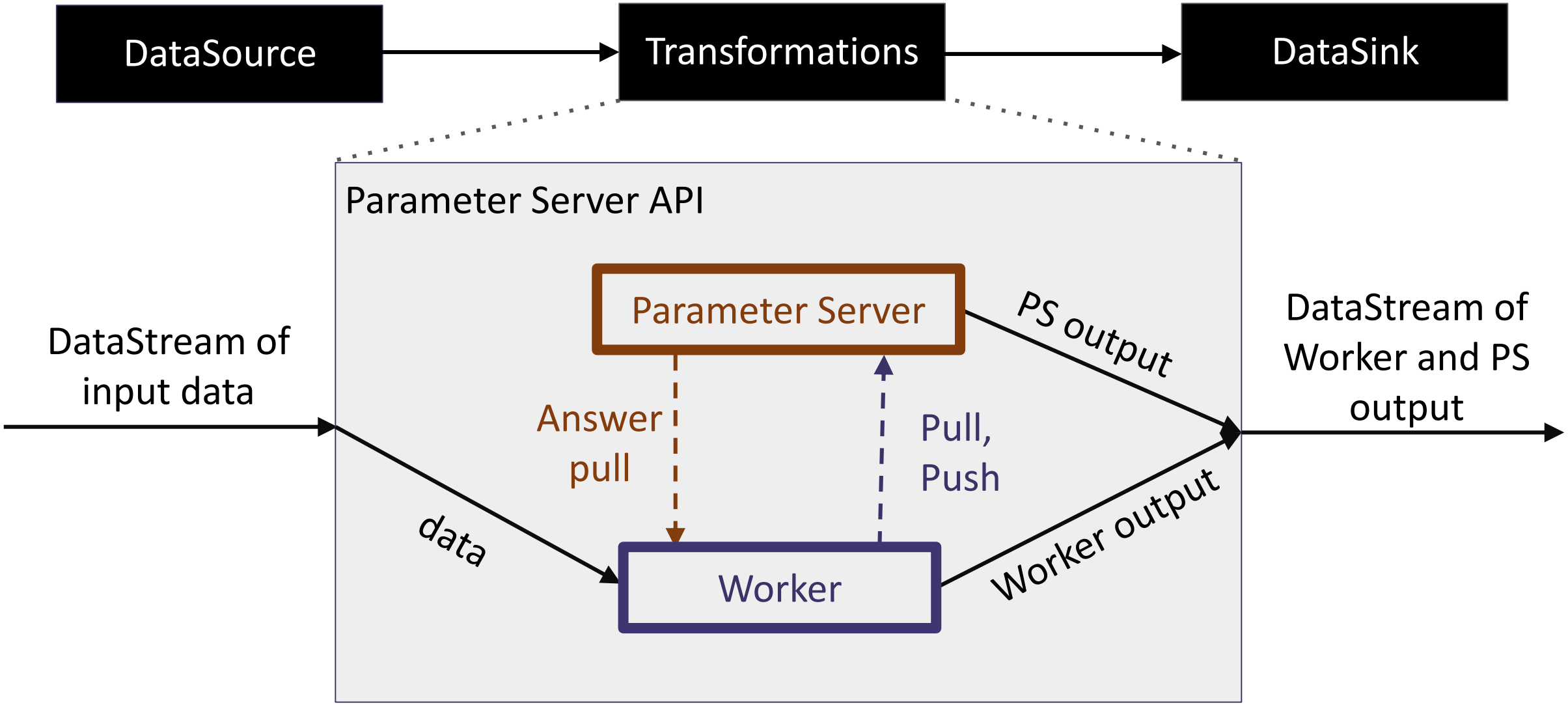
Implementation?



Implementation?

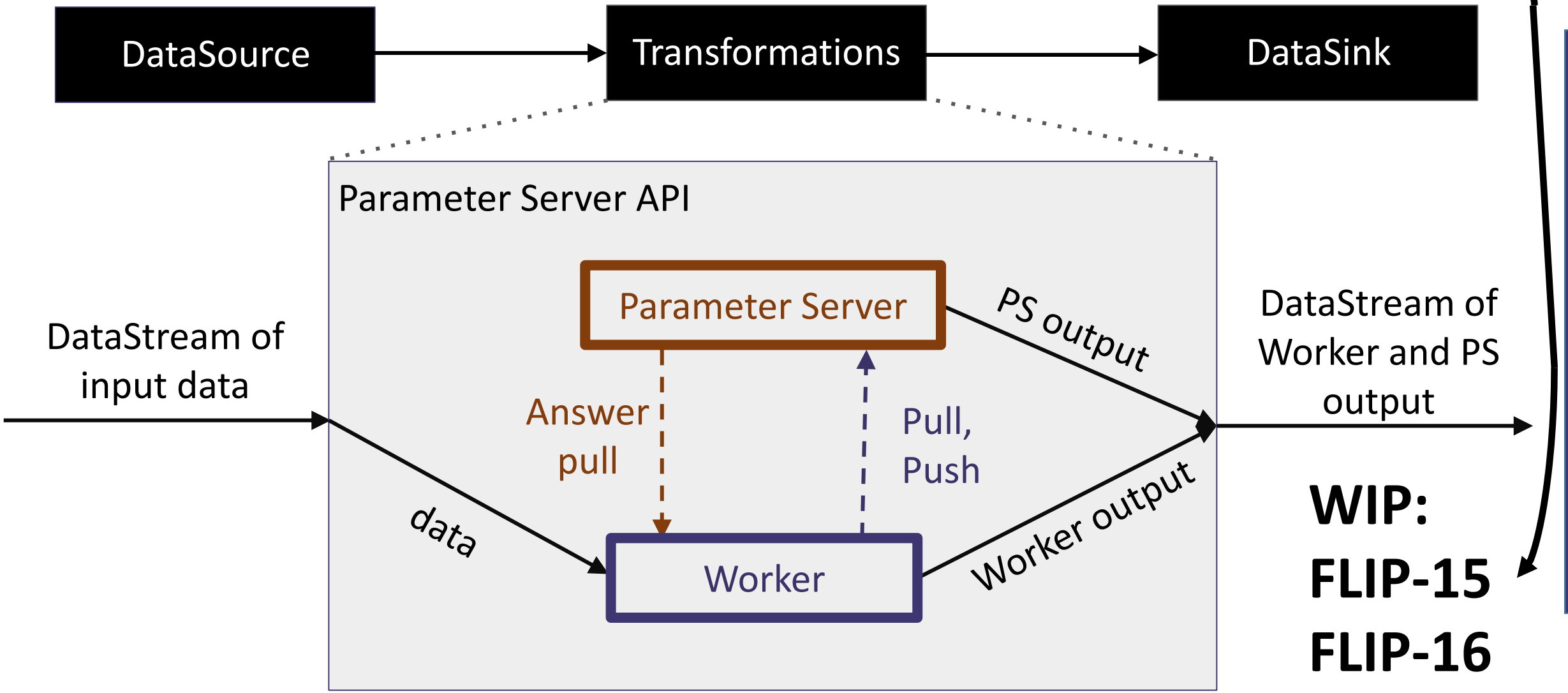


Implementation: **Loops API**



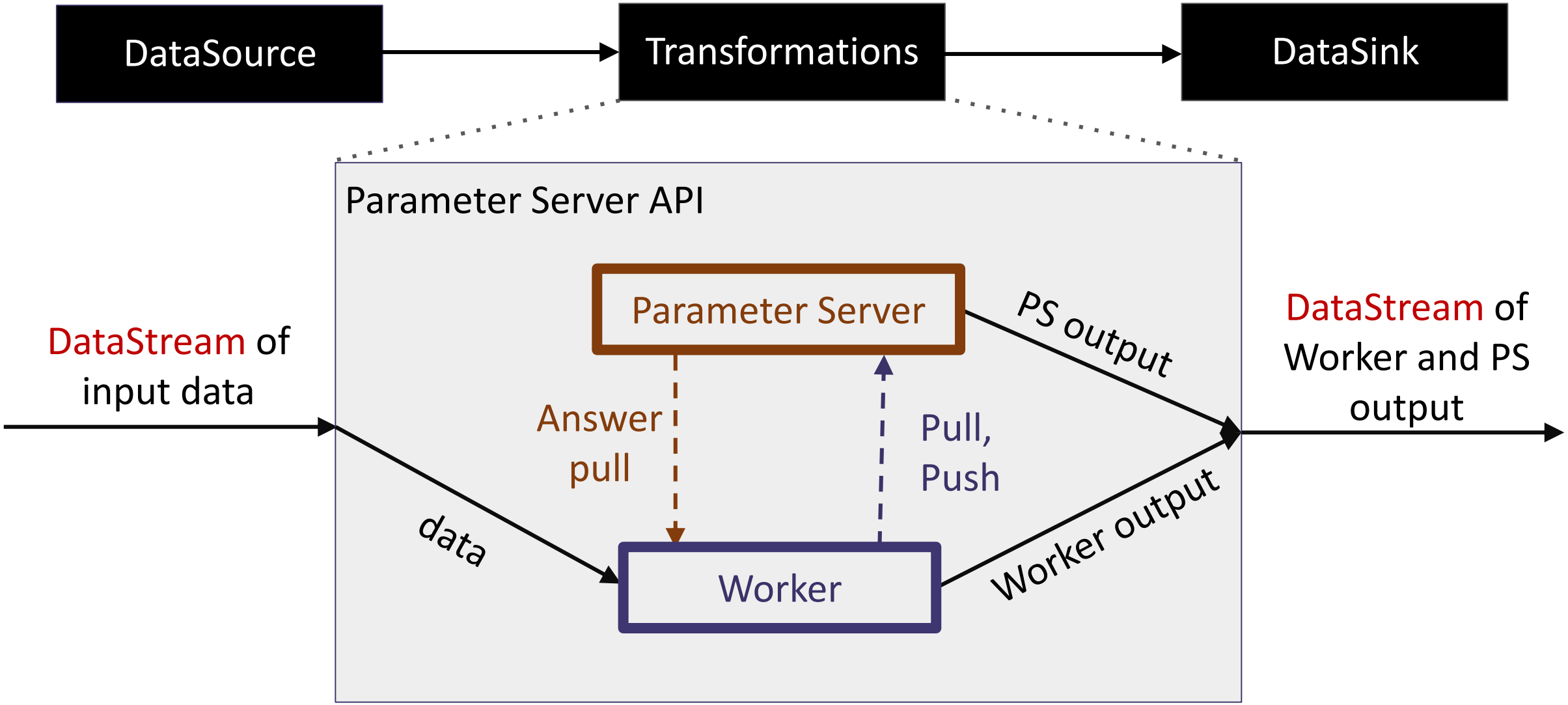
Implementation: **Loops API**

NOT MATURE

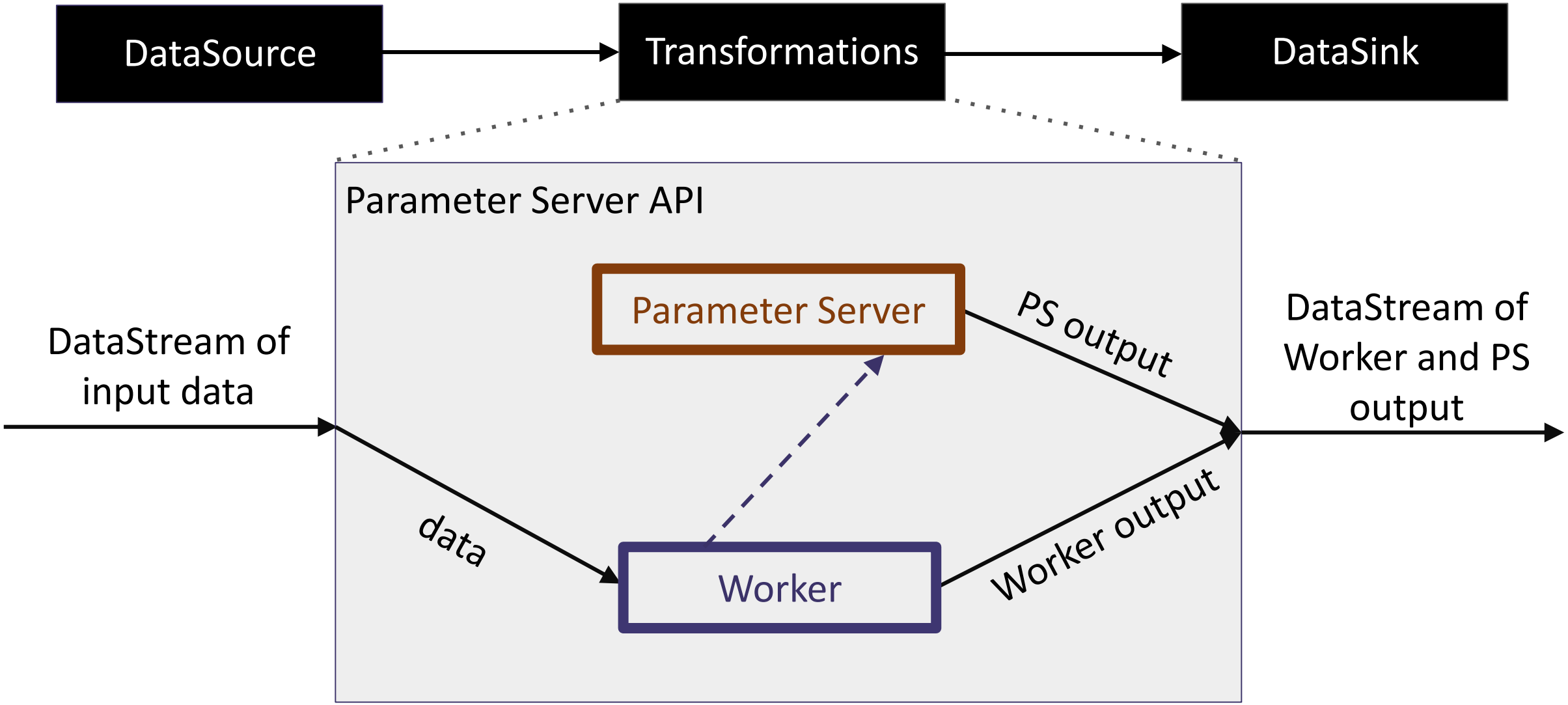


WIP:
FLIP-15
FLIP-16

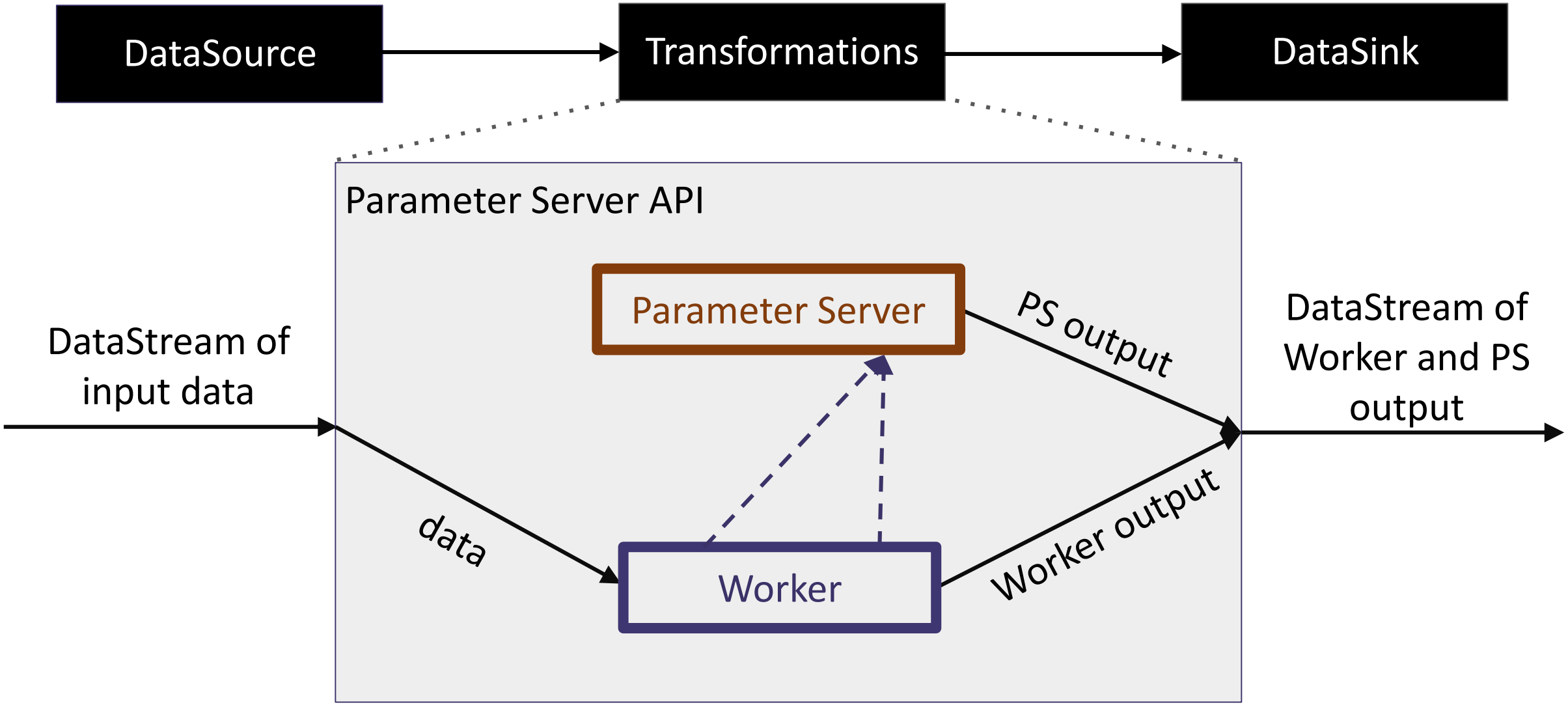
Implementation



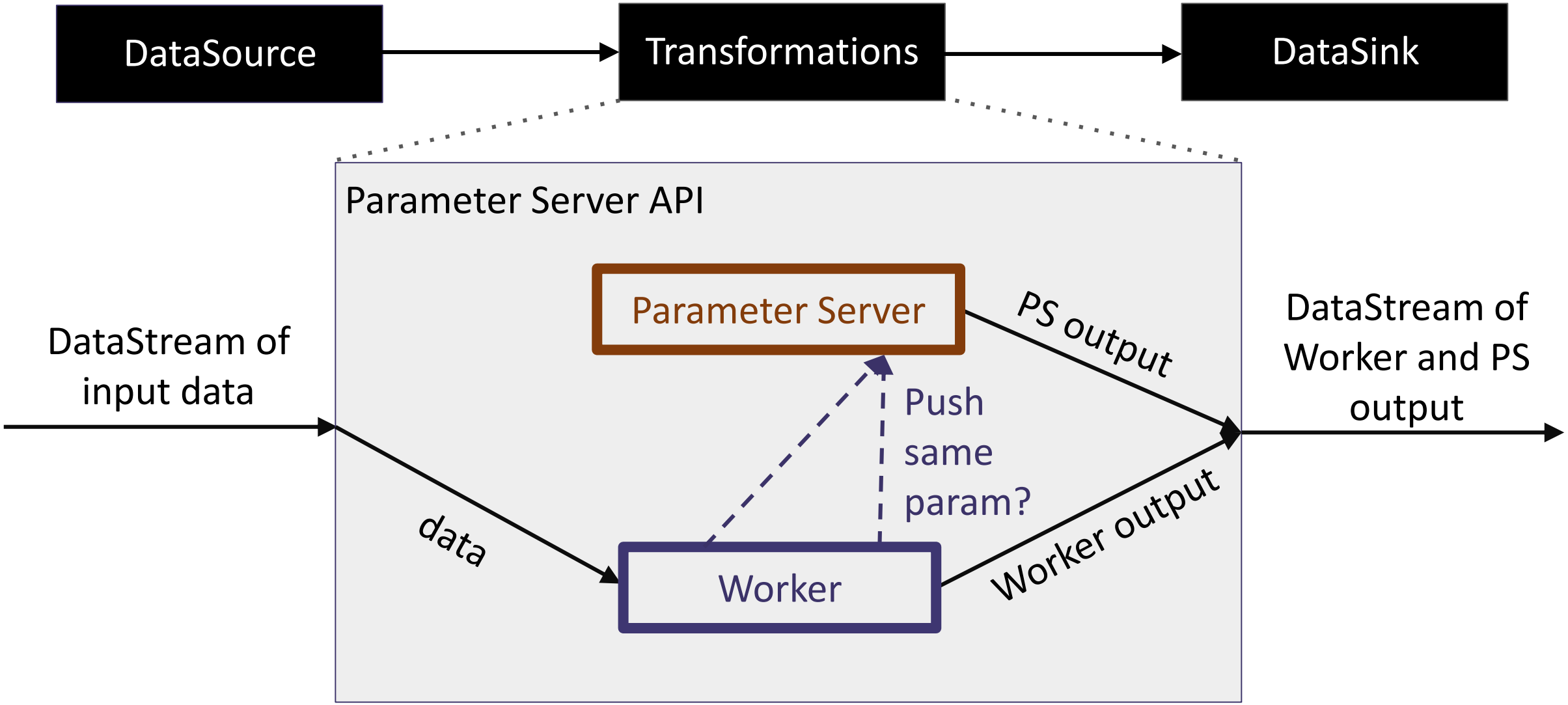
Implementation



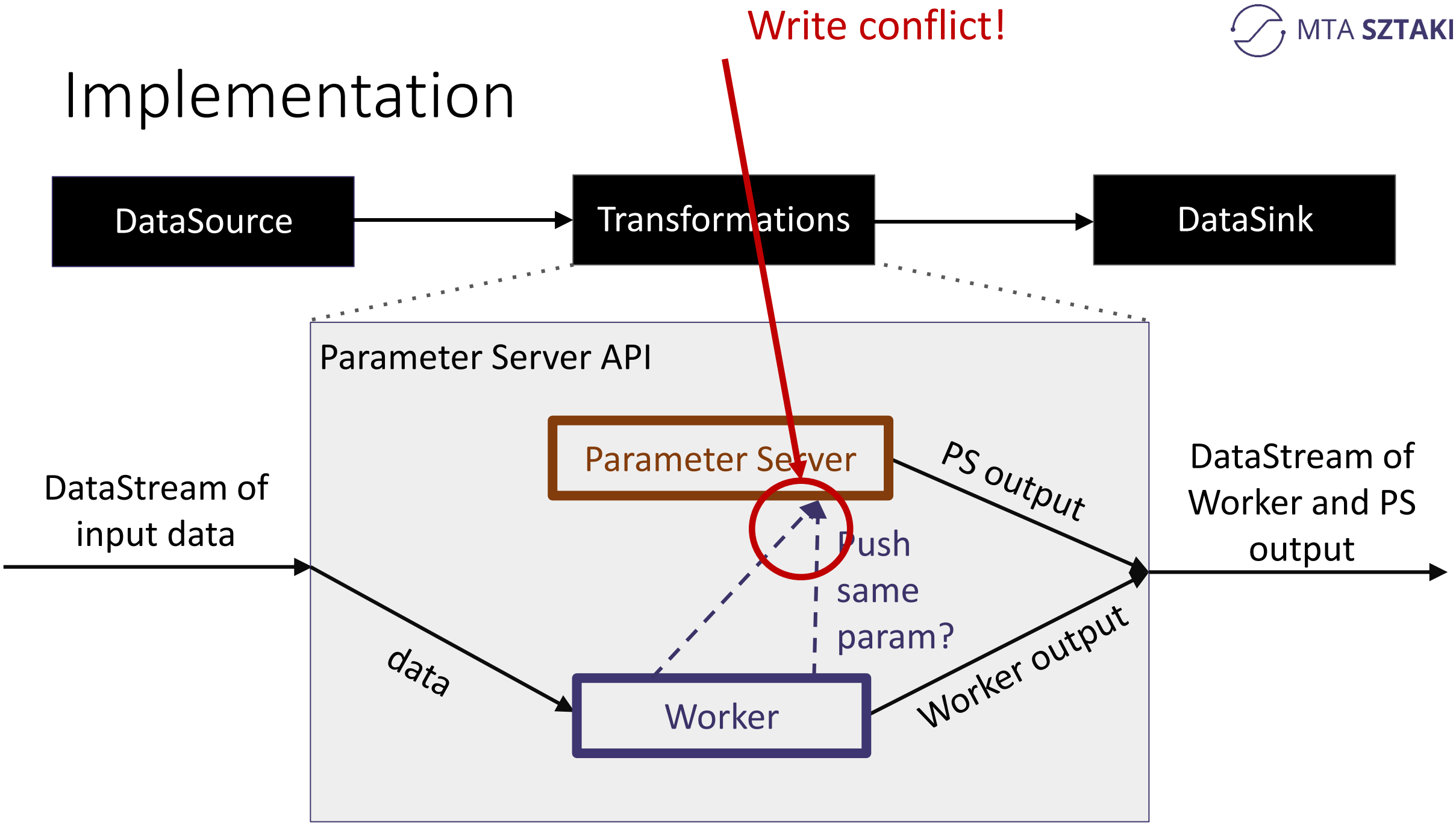
Implementation



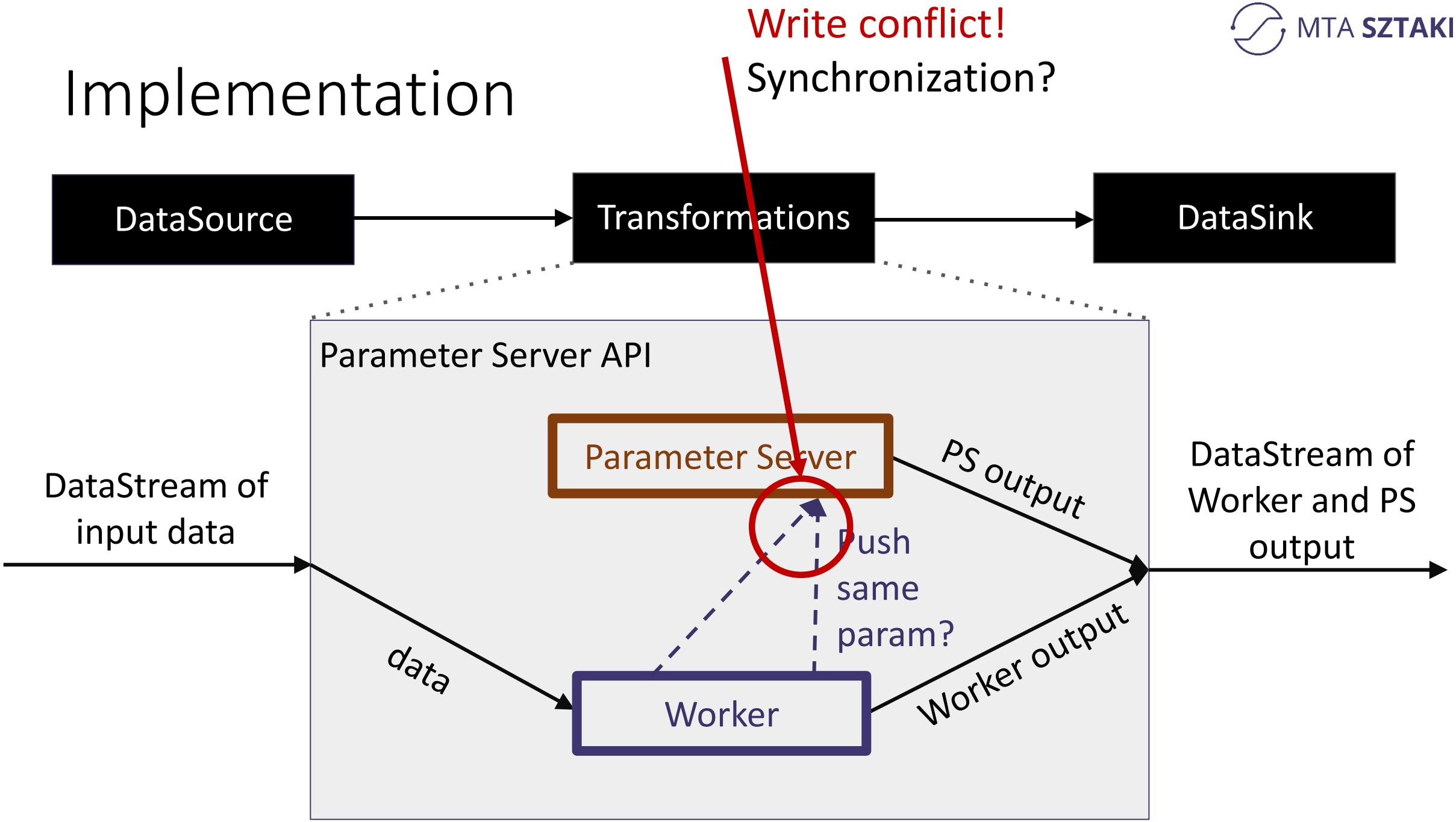
Implementation



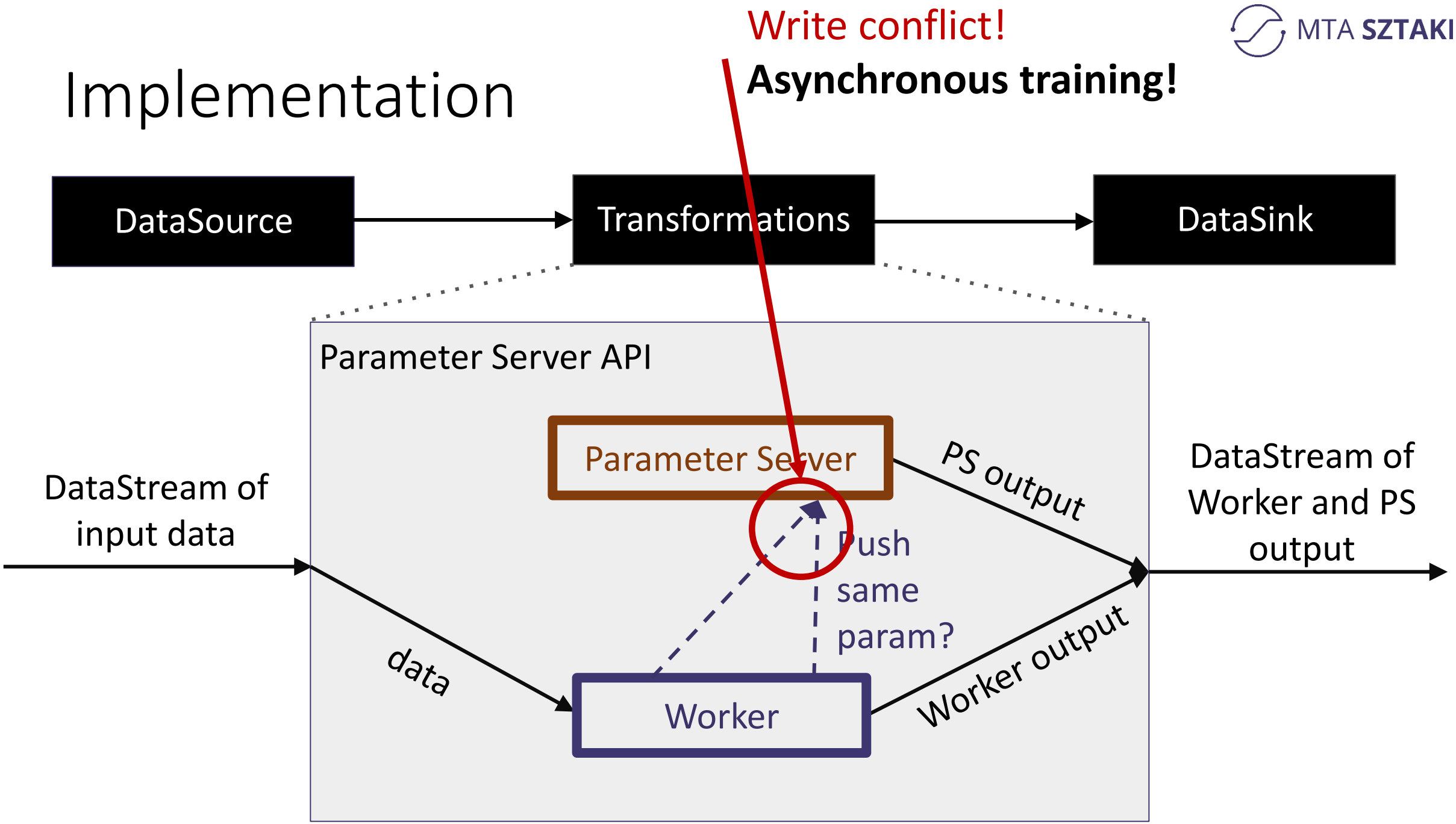
Implementation

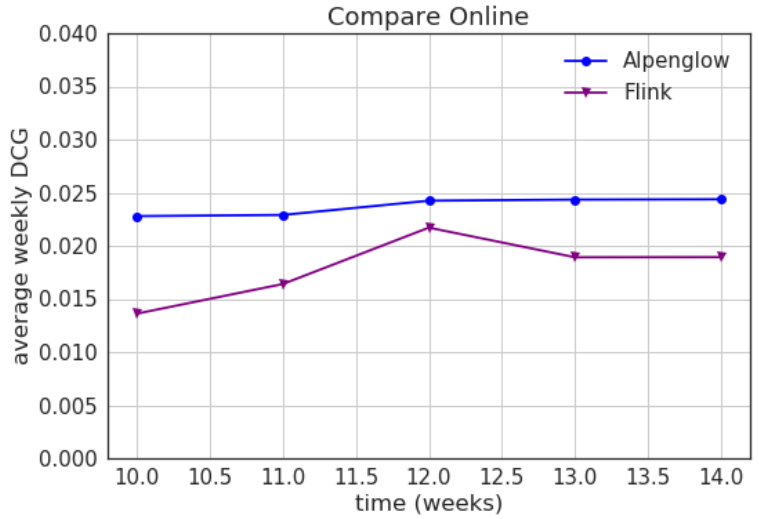


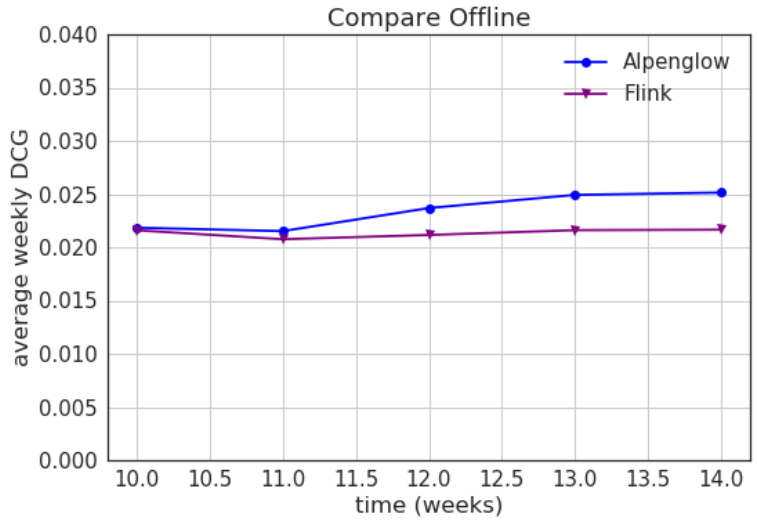
Implementation

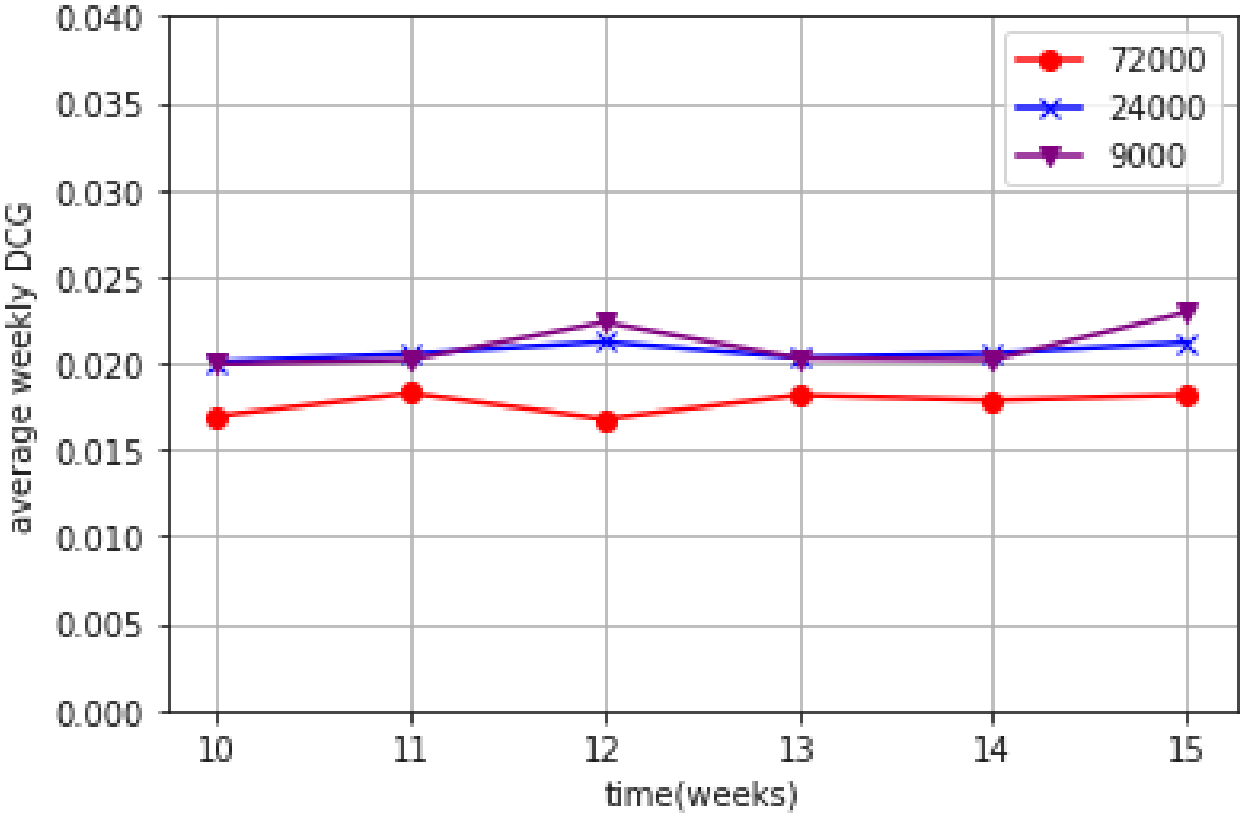


Implementation




Compare Online
Alpenglow
Flink
average weekly DCG
time (weeks)
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0


Compare Offline
Alpenglow
Flink
average weekly DCG
time (weeks)
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
10.0
10.5
11.0
11.5
12.0
12.5
13.0
13.5
14.0


72000
24000
9000
average weekly DCG
time(weeks)
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
10
11
12
13
14
15

MTA SZTAKI

Framework and library

- Framework
  - Easy to implement new algorithms
- Library
  - Matrix Factorization
  - Factorization Machine
  - Passive Aggressive
  - Sketch

Thank you for your attention

Source code:
https://github.com/FlinkML/flink-parameter-server

Dániel Berecz

@DBerecz

bdaniel@info.ilab.sztaki.hu

Gábor Hermann
mail@gaborhermann.com

https://github.com/rpalovics/Alpenglow

M. Li, et al.: "Scaling Distributed Machine Learning with the Parameter Server" 2014.

K. Crammer, et al.: “Online Passive-Aggressive Algorithms” 2006.

S. Schelter, et al.: "Factorbird - A Parameter Server Approach to Distributed Matrix Factorization." 2014.

R. Gemulla, et al. "Large-scale matrix factorization with distributed stochastic gradient descent” 2011.

Backups

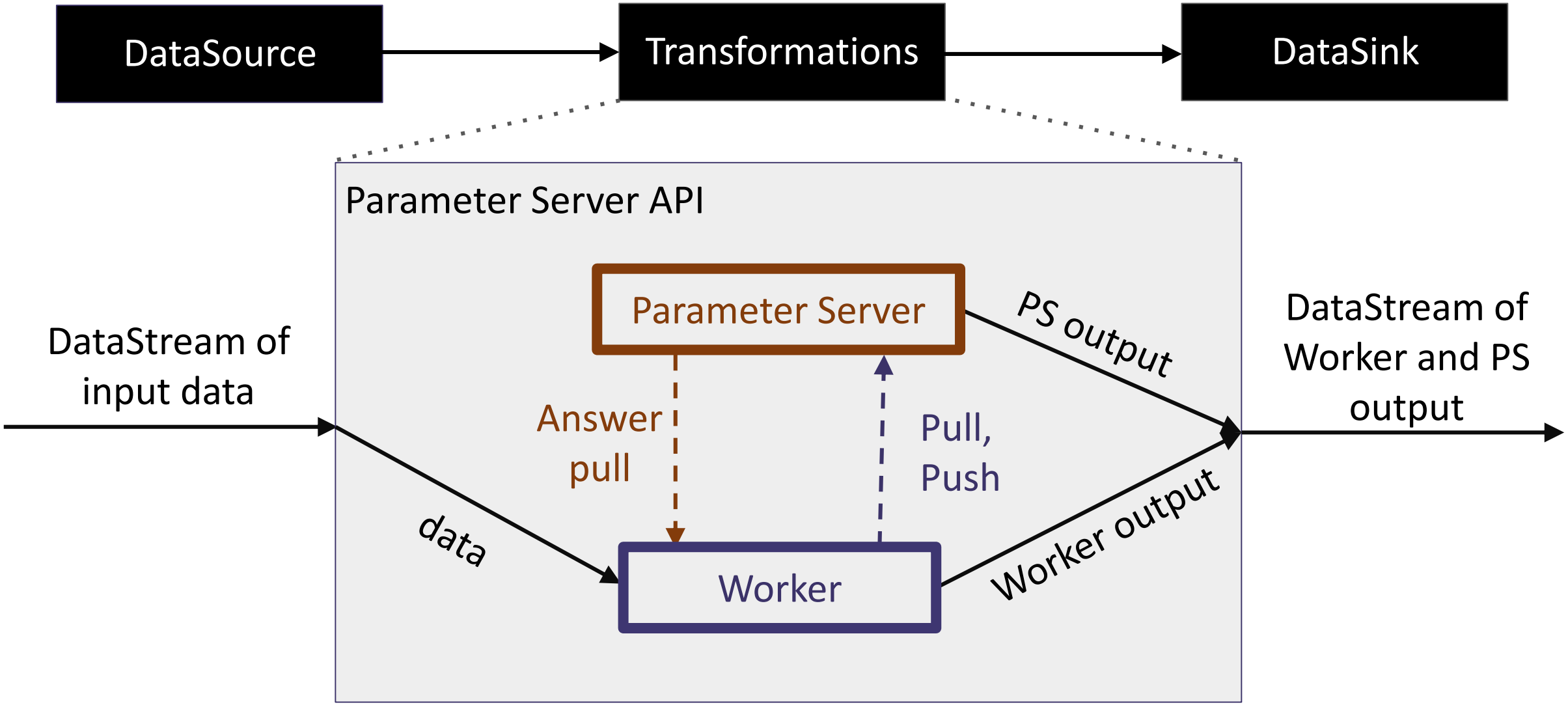
Batch? Streaming?

Batch vs streaming?

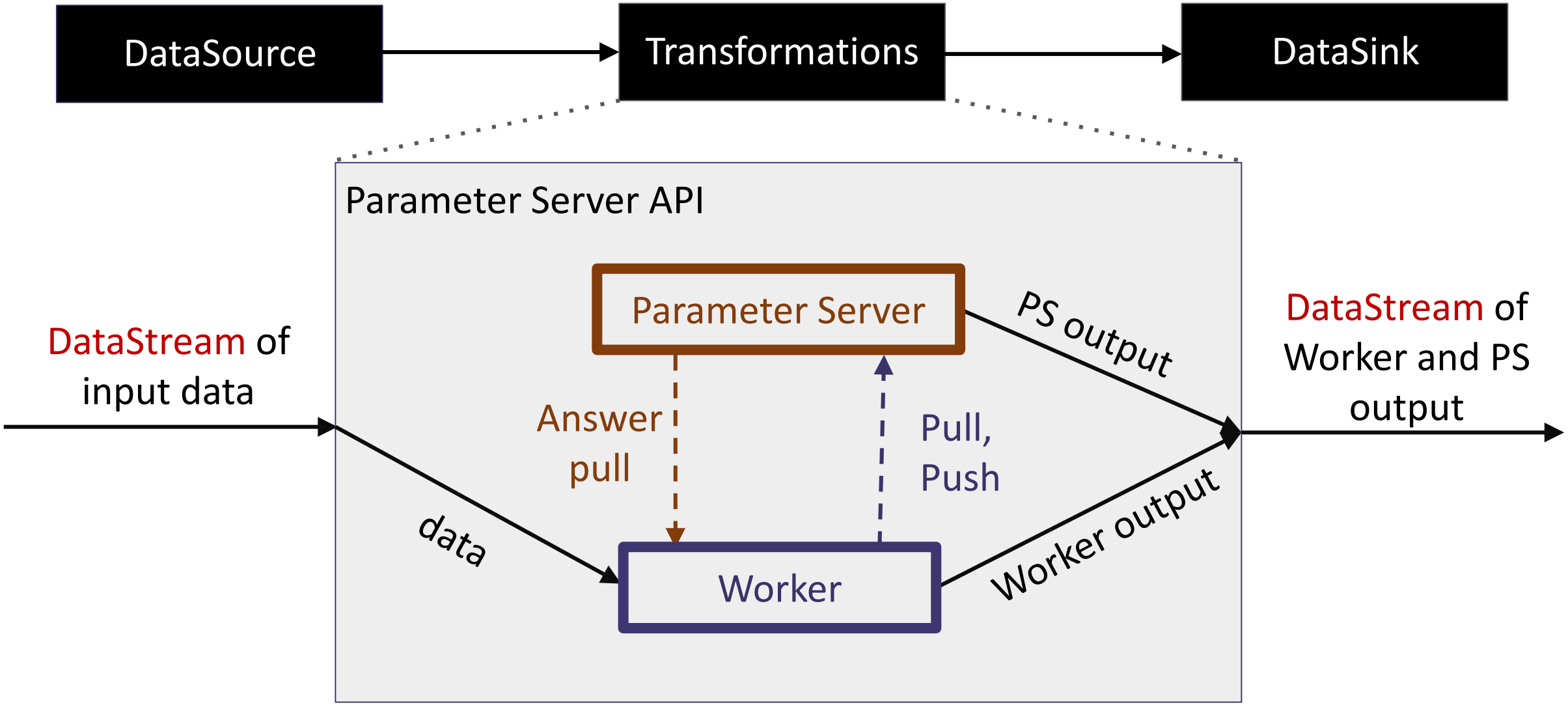
Batch vs streaming?
Offline vs online?

MTA SZTAKI

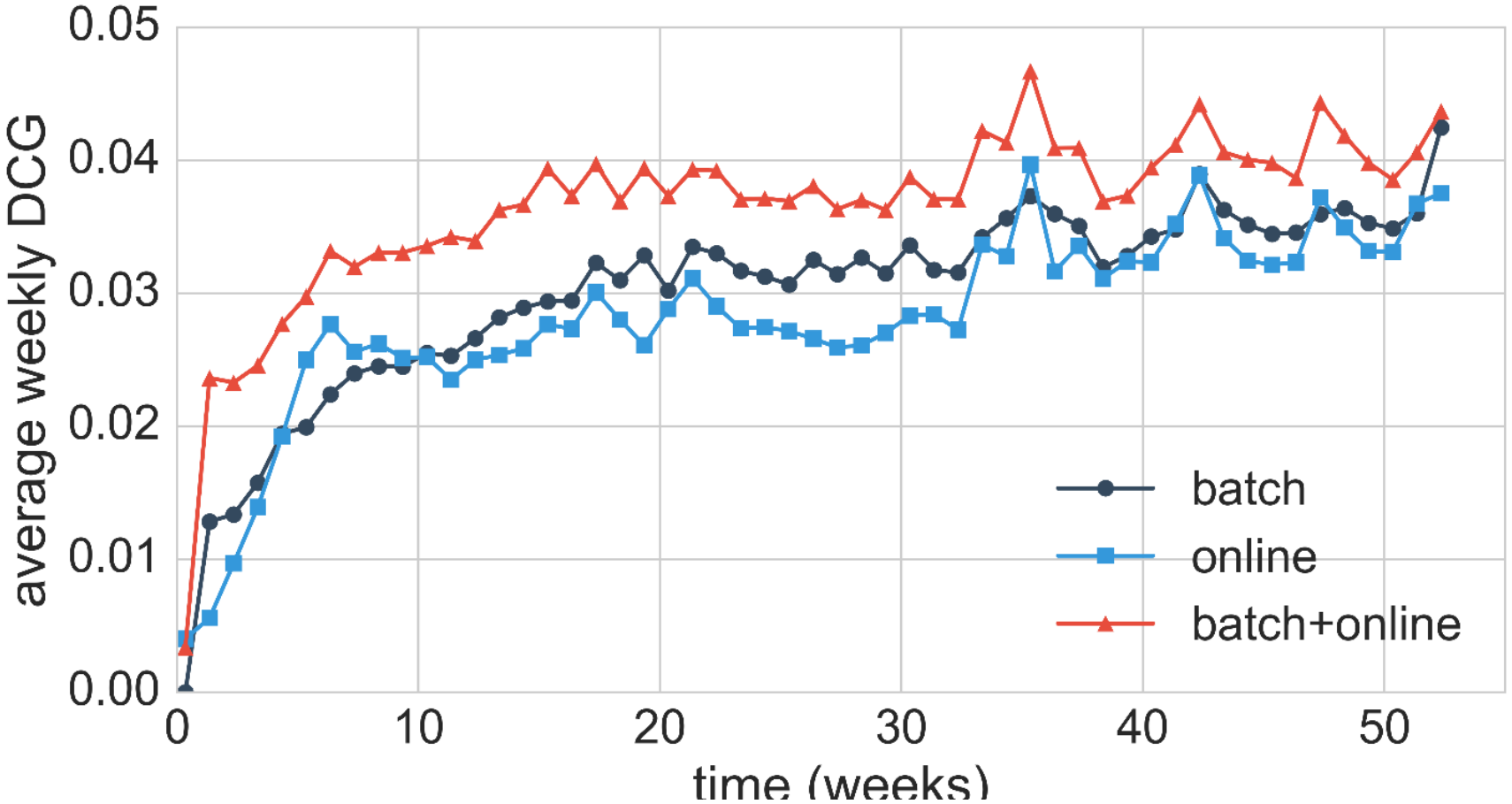
Online on streaming



Online on streaming

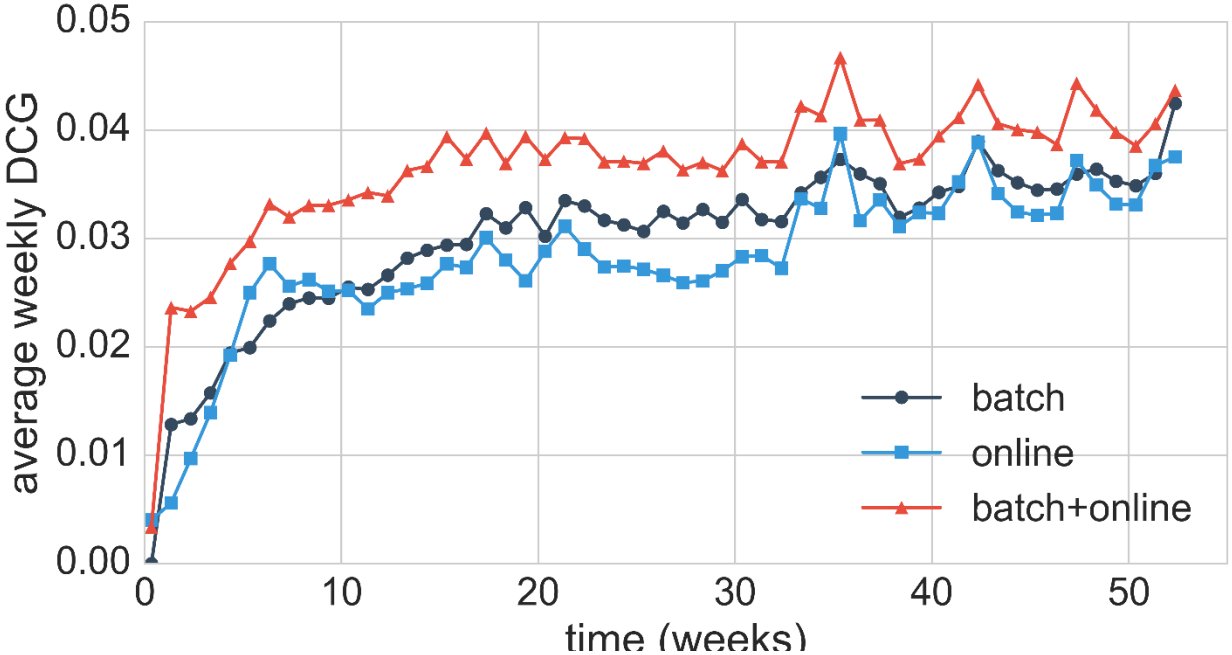


Batch + online combination



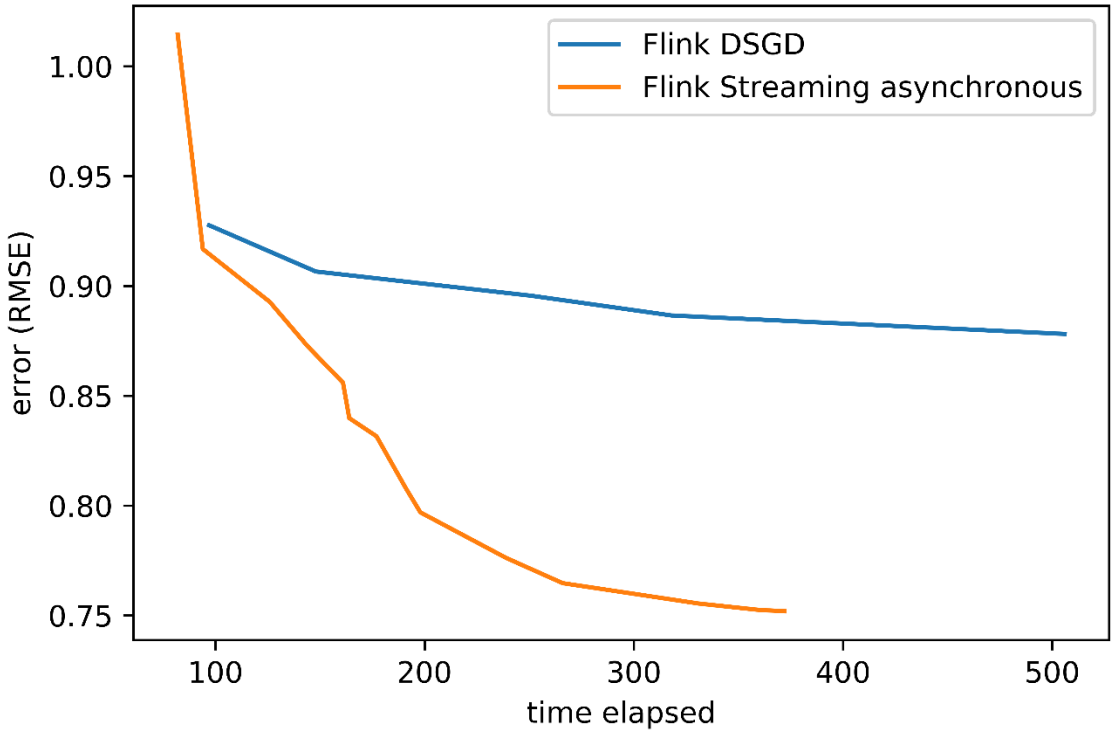
Batch + online combination

- 30M music listening Last.fm dataset
- Weekly batch training
- Evaluation weekly average
  - on every incoming listening
- Around 45.000 users



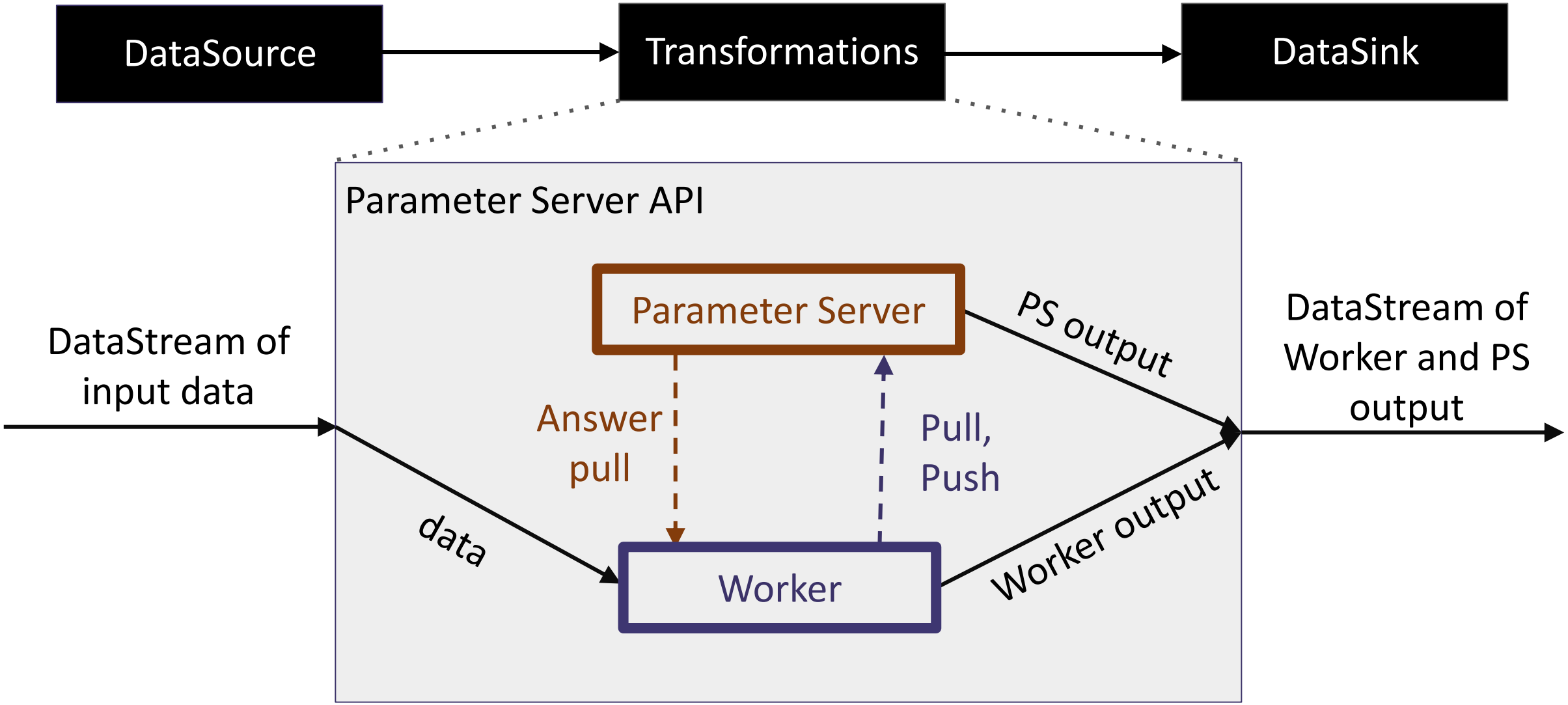
Batch on Flink Streaming

- Movielens 1M movie rating dataset
- Using 6 nodes, 4 cores each



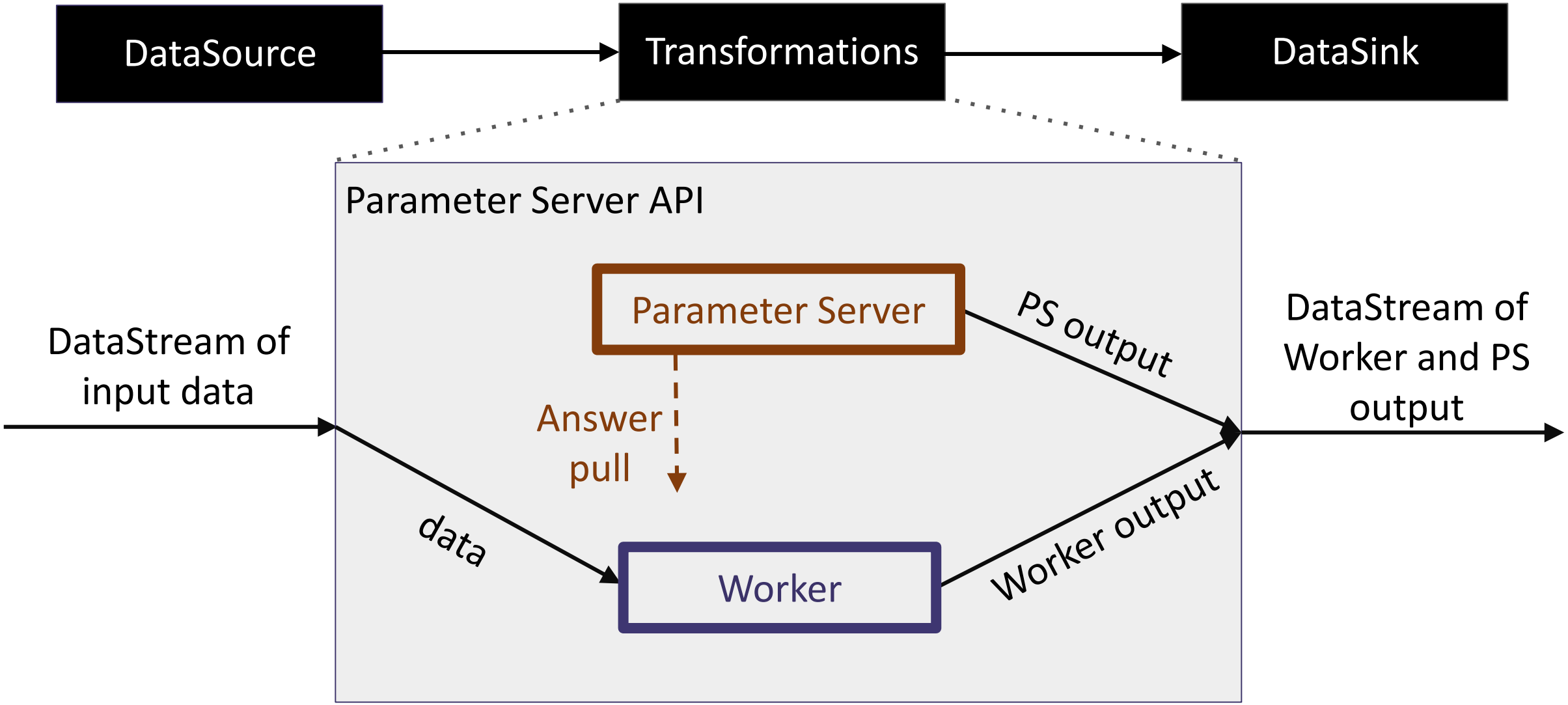
Implementation: **Loops API**

NOT MATURE



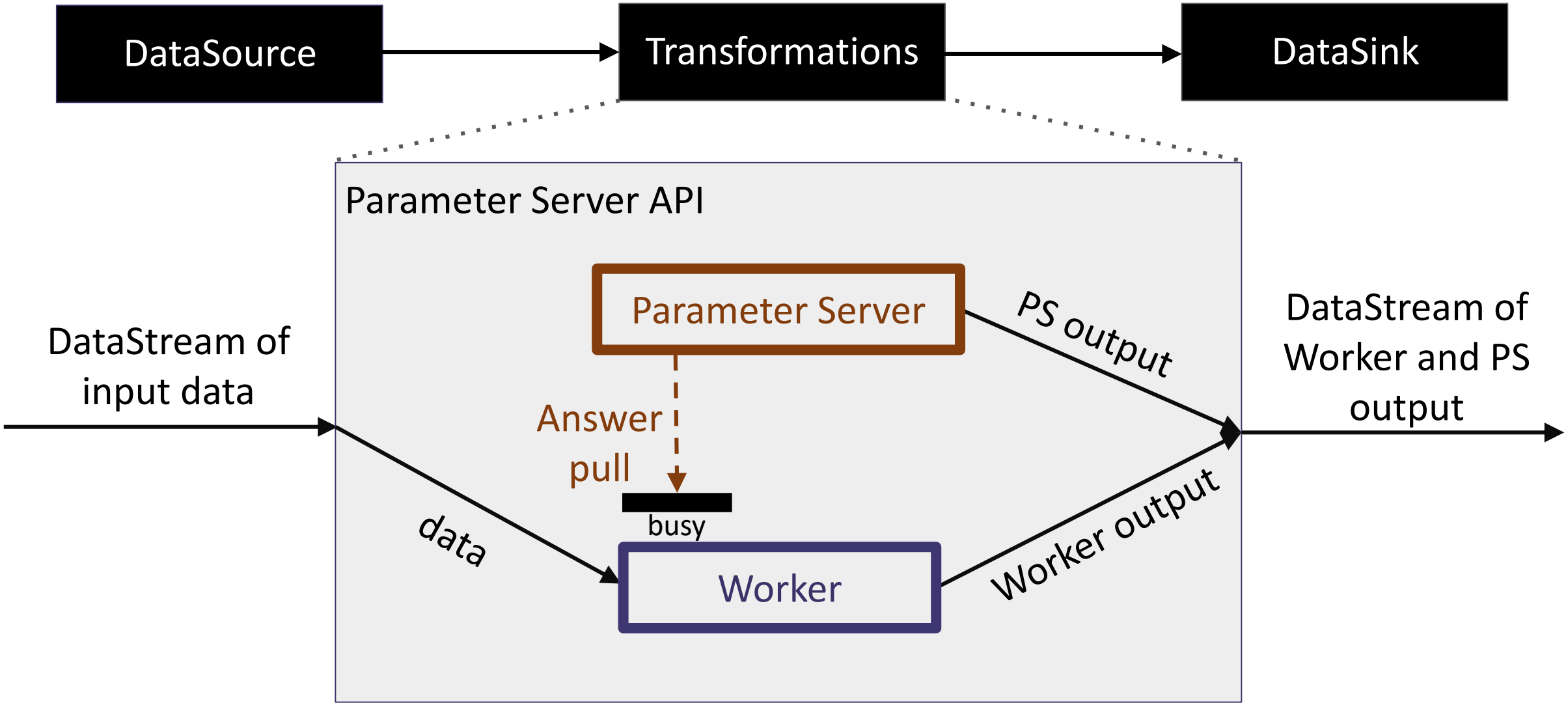
Implementation: **Loops API**

NOT MATURE



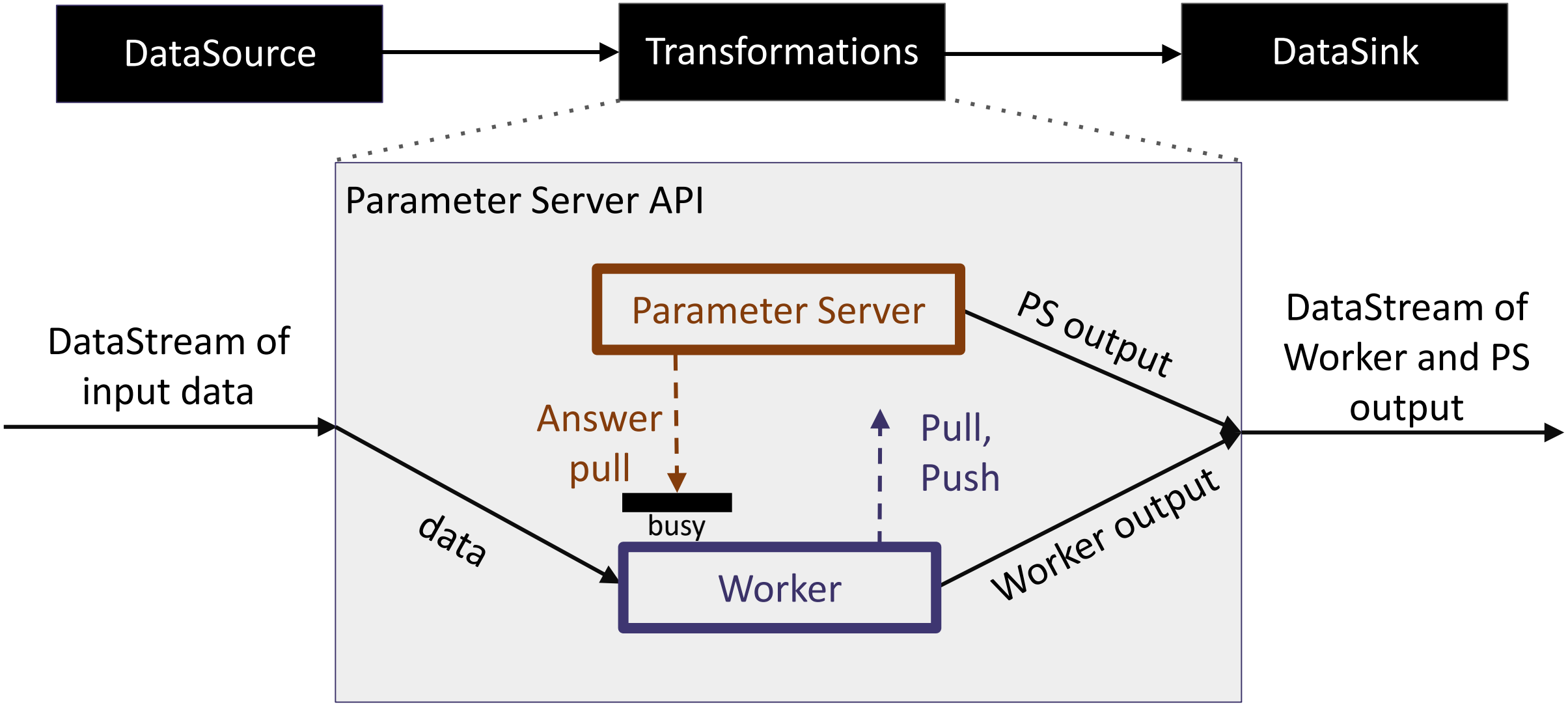
Implementation: **Loops API**

NOT MATURE



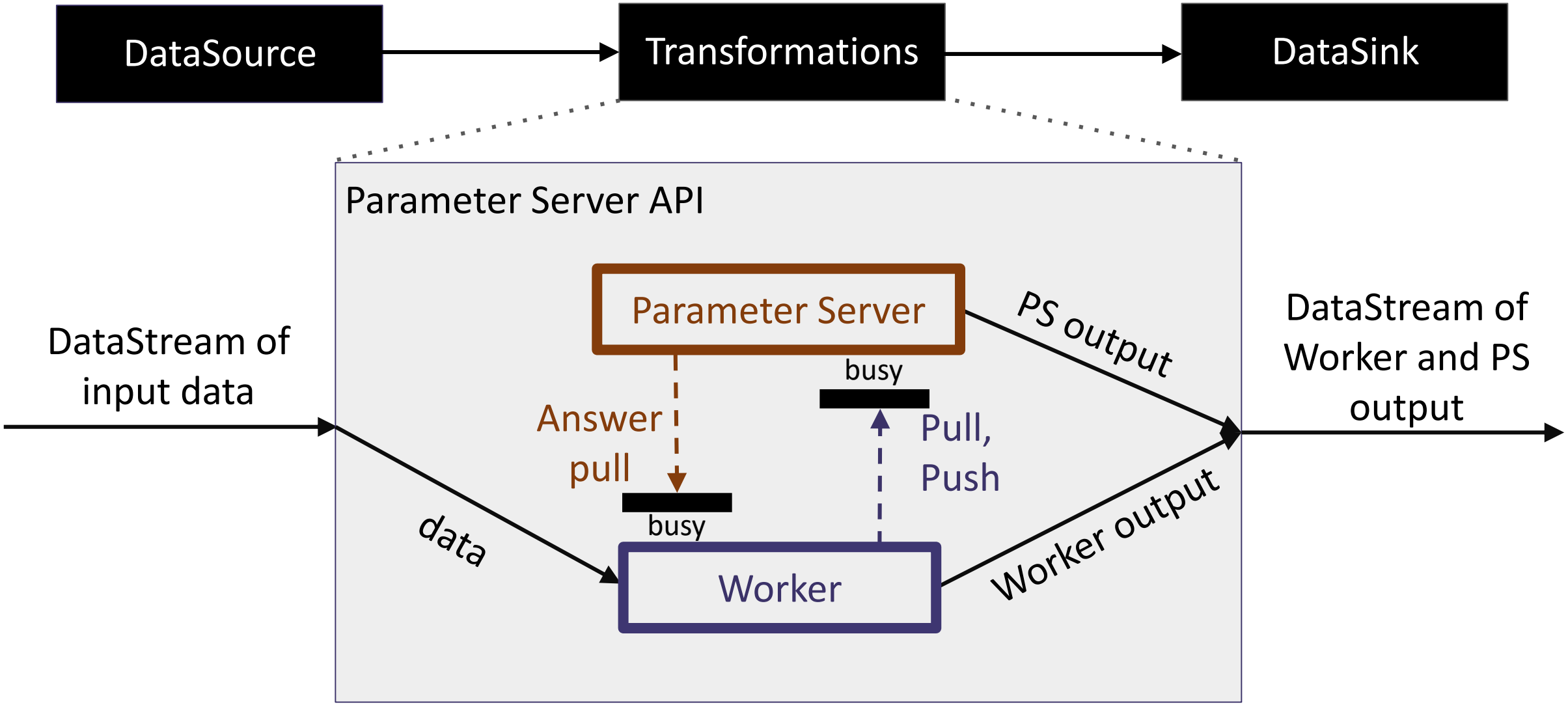
Implementation: **Loops API**

NOT MATURE



Implementation: **Loops API**

NOT MATURE



Implementation: **Loops API**

NOT MATURE