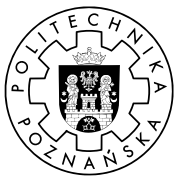
Having Your Cake and Eating it Too: Combining Strong and Eventual Consistency

Konrad Siek and Paweł T. Wojciechowski

Poznań University of Technology

{konrad.siek,pawel.t.wojciechowski}@cs.put.edu.pl

13 IV 2013



Distributed Systems Group

http://dsg.cs.put.poznan.pl

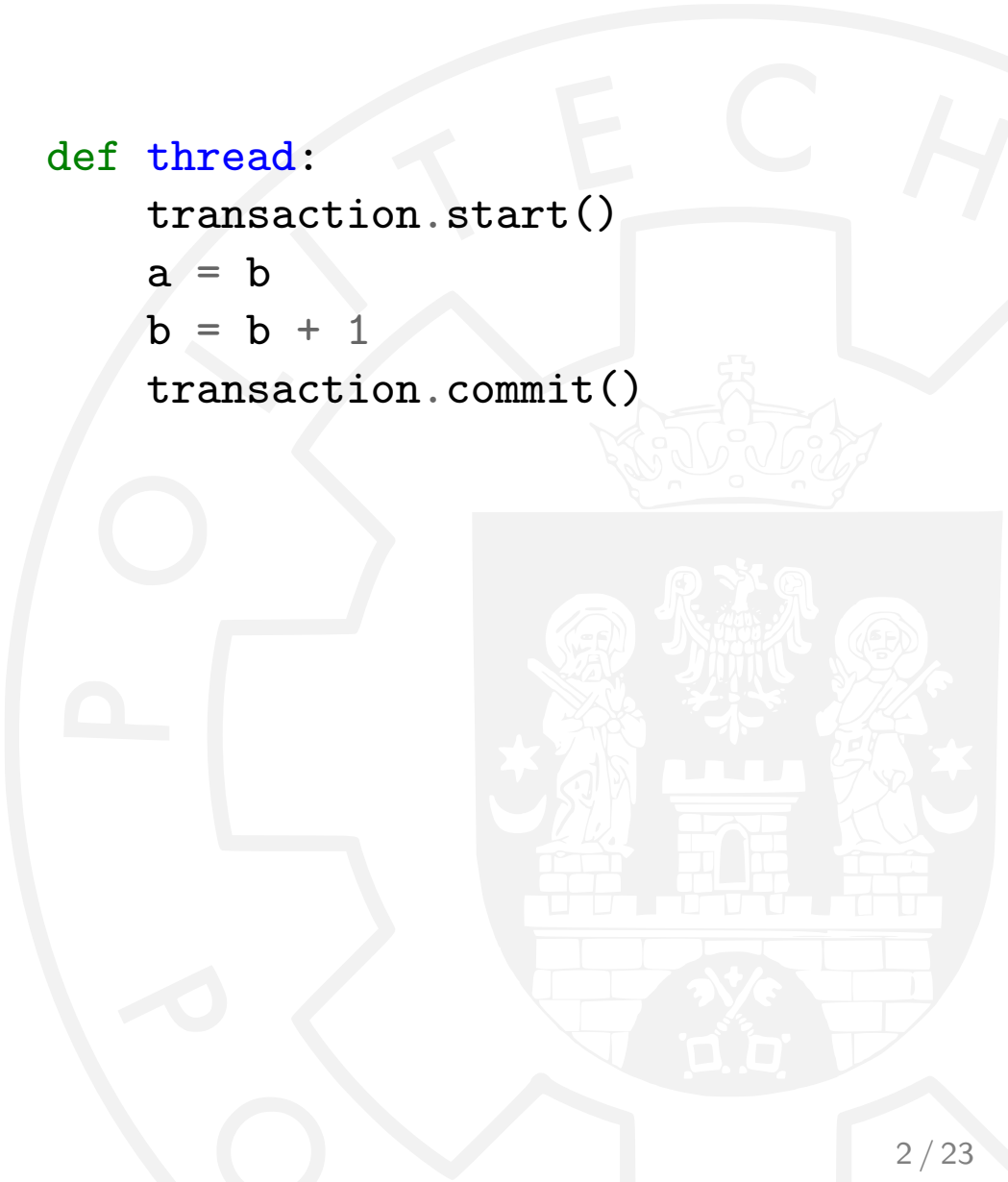
Software Transactional Memory

```
def thread:
    lock_a.acquire()
    lock_b.acquire()
    a = b
    lock_a.release()
    b = b + 1
    lock_b.release()
```

Software Transactional Memory

```
def thread:
    lock_a.acquire()
    lock_b.acquire()
    a = b
    lock_a.release()
    b = b + 1
    lock_b.release()
```

```
def thread:
    transaction.start()
    a = b
    b = b + 1
    transaction.commit()
```



Software Transactional Memory

```
def thread:
    lock_a.acquire()
    lock_b.acquire()
    a = b
    lock_a.release()
    b = b + 1
    lock_b.release()
```

```
def thread:
    transaction.start()
    a = b
    b = b + 1
    transaction.commit()
```

Advantages:

- ease of use on top
- efficient concurrency control under the hood

Transaction Abstraction

Transaction:

$$T_i \; [\![\; op_1, \; op_2, \; ..., \; op_n \;]\!]$$

where $op = \{ \; r(x)v, \; w(x)v, \; ... \; \}$

and x is some shared object

Commitment:

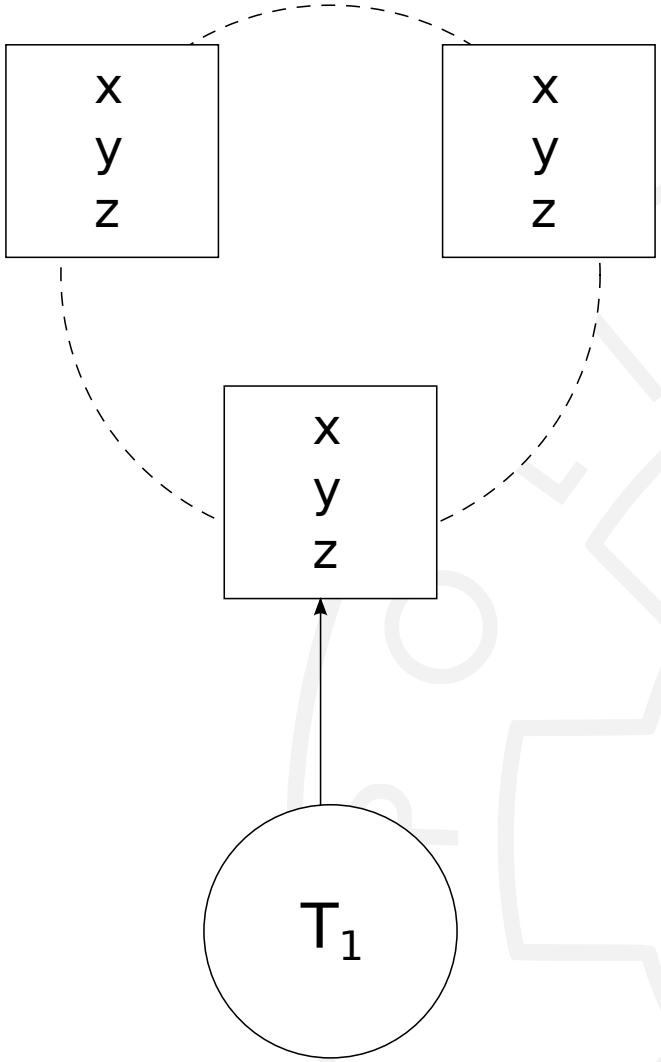
$$\{x = 1\} \quad T_i \; [\![\; w(x)2 \;]\!] \quad \{x = 2\}$$

Rollback:

$$\{x = 1\} \quad T_i \; [\![\; w(x)2, \quad \hookleftarrow \quad \{x = 1\}$$

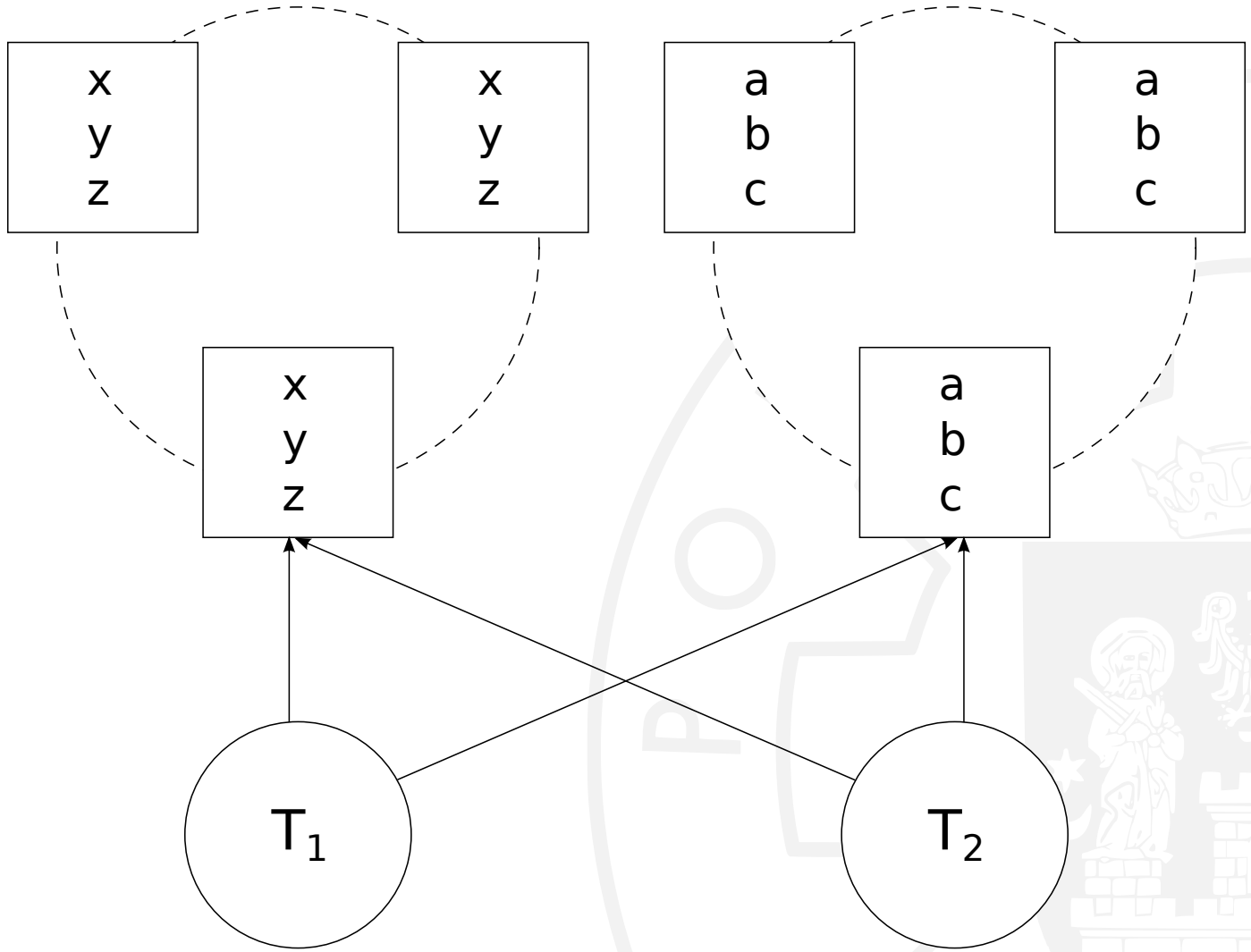
$$\{x = 1\} \quad T_i \; [\![\; w(x)2, \quad \hookleftarrow \quad \rightarrow \quad T_i' \; [\![\; w(x)2 \;]\!] \quad \{x = 2\}$$

Distributed Transactional Memory



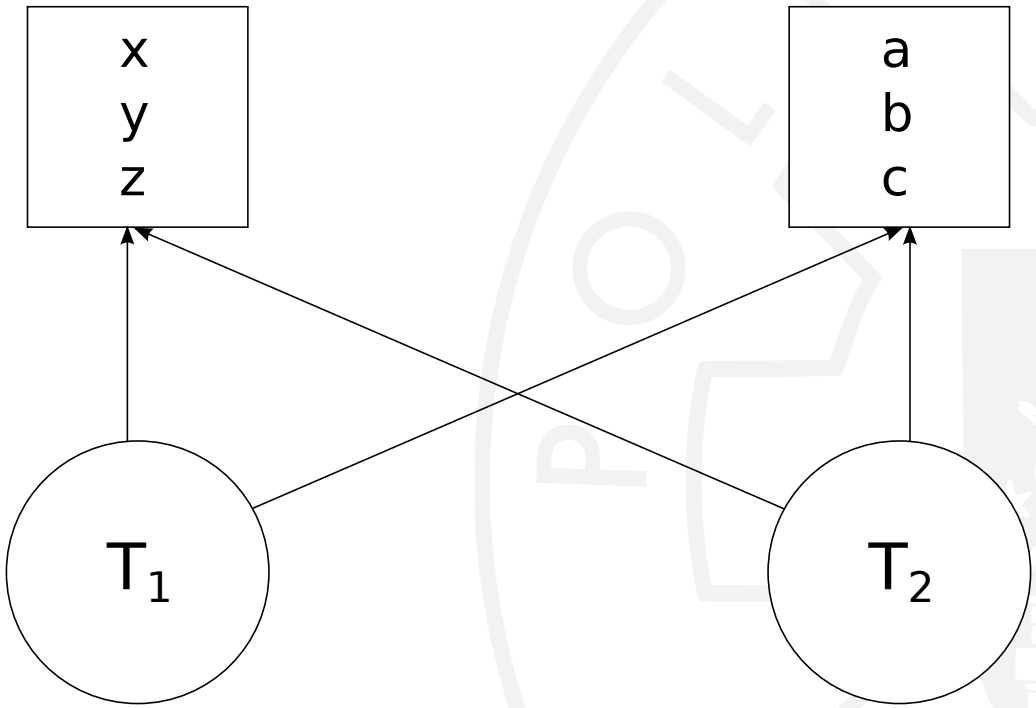
Replicated TM

Distributed Transactional Memory



Distributed Transactions

Distributed Transactional Memory



Distributed Transactions

Supremum Versioning Algorithm

Pessimistic approach

$$\{x=1, y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3, y=2\}$$

Supremum Versioning Algorithm

Pessimistic approach

$$\{x = 1, y = 1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!]$$
$$| \; T_2 \; [\![\qquad\qquad \searrow r(x)2, w(x)3 \;]\!] \quad \{x = 3, y = 2\}$$

- Defer execution to prevent conflicts

Supremum Versioning Algorithm

Pessimistic approach

$$\{x = 1, y = 1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!] \; | \; T_2 \; [\![\quad \searrow r(x)2, w(x)3 \;]\!] \quad \{x = 3, y = 2\}$$

- Defer execution to prevent conflicts (tolerate high contention)

Supremum Versioning Algorithm

Pessimistic approach

$$\{x = 1, y = 1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!]$$
$$\mid T_2 \; [\![\qquad \searrow r(x)2, w(x)3 \;]\!] \quad \{x = 3, y = 2\}$$

- Defer execution to prevent conflicts (tolerate high contention)
- Avoid (most) forced aborts

Supremum Versioning Algorithm

Pessimistic approach

$$\{x = 1, y = 1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!]$$
$$\mid T_2 \; [\![\qquad \searrow r(x)2, w(x)3 \;]\!] \quad \{x = 3, y = 2\}$$

- Defer execution to prevent conflicts (tolerate high contention)
- Avoid (most) forced aborts (safe irrevocable operations)

Supremum Versioning Algorithm

Pessimistic approach

$$\{x=1, y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!] \quad | \; T_2 \; [\![\; \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3, y=2\}$$

- Defer execution to prevent conflicts (tolerate high contention)
- Avoid (most) forced aborts (safe irrevocable operations)

Early release on last use

$$\{x=1, y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2, r(y)1, w(y)2 \;]\!] \quad | \; T_2 \; [\![\; \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3, y=2\}$$

Supremum Versioning Algorithm

Pessimistic approach

$$\{x=1,y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!] \; | \; T_2 \; [\![\; \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3,y=2\}$$

- Defer execution to prevent conflicts (tolerate high contention)
- Avoid (most) forced aborts (safe irrevocable operations)

Early release on last use

$$\{x=1,y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2, r(y)1, w(y)2 \;]\!] \; | \; T_2 \; [\![\; \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3,y=2\}$$

Supremum Versioning Algorithm

Pessimistic approach

$$\{x=1, y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2 \;]\!]$$
$$\mid T_2 \; [\![\qquad \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3, y=2\}$$

- Defer execution to prevent conflicts (tolerate high contention)
- Avoid (most) forced aborts (safe irrevocable operations)

Early release on last use

$$\{x=1, y=1\} \quad T_1 \; [\![\; r(x)1, w(x)2, r(y)1, w(y)2 \;]\!]$$
$$\mid T_2 \; [\![\qquad \searrow r(x)2, w(x)3 \;]\!] \quad \{x=3, y=2\}$$

Completely distributed (no leader, dispatcher, etc.)

Supremum Versioning Algorithm

Pessimistic approach

$$\{x=1,y=1\}\quad T_1\ [\![\ r(x)1, w(x)2\ ]\!] \\ \mid T_2\ [\![\qquad \searrow r(x)2, w(x)3\ ]\!]\quad \{x=3,y=2\}$$

- Defer execution to prevent conflicts (tolerate high contention)
- Avoid (most) forced aborts (safe irrevocable operations)

Early release on last use

$$\{x=1,y=1\}\quad T_1\ [\![\ r(x)1, w(x)2, r(y)1, w(y)2\ ]\!] \\ \mid T_2\ [\![\qquad \searrow r(x)2, w(x)3\ ]\!]\quad \{x=3,y=2\}$$

Completely distributed (no leader, dispatcher, etc.)

Strong consistency

Bank Application

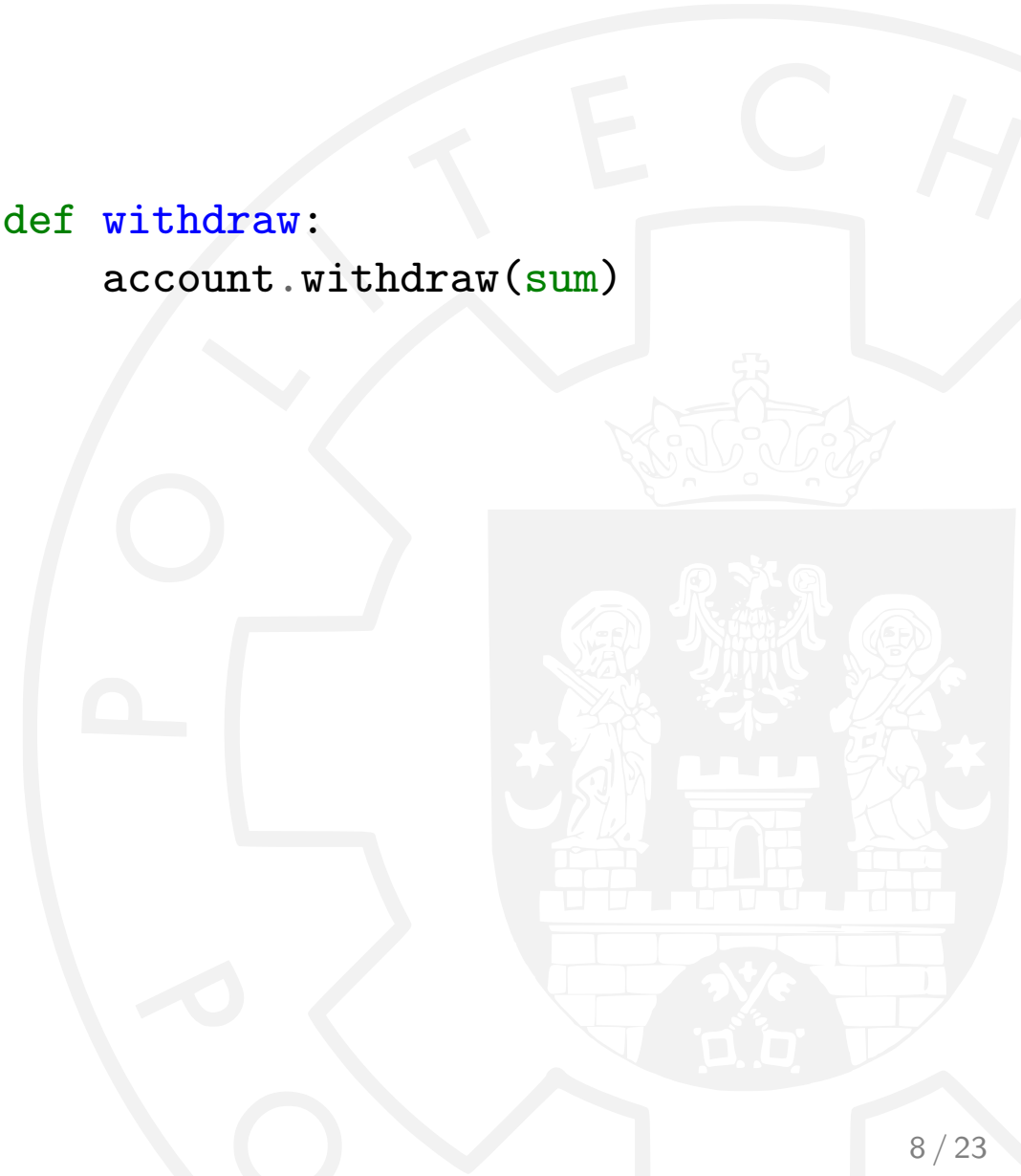
Bank Application

```
def deposit:
    account.deposit(sum)
```

Bank Application

```
def deposit:
    account.deposit(sum)
```

```
def withdraw:
    account.withdraw(sum)
```

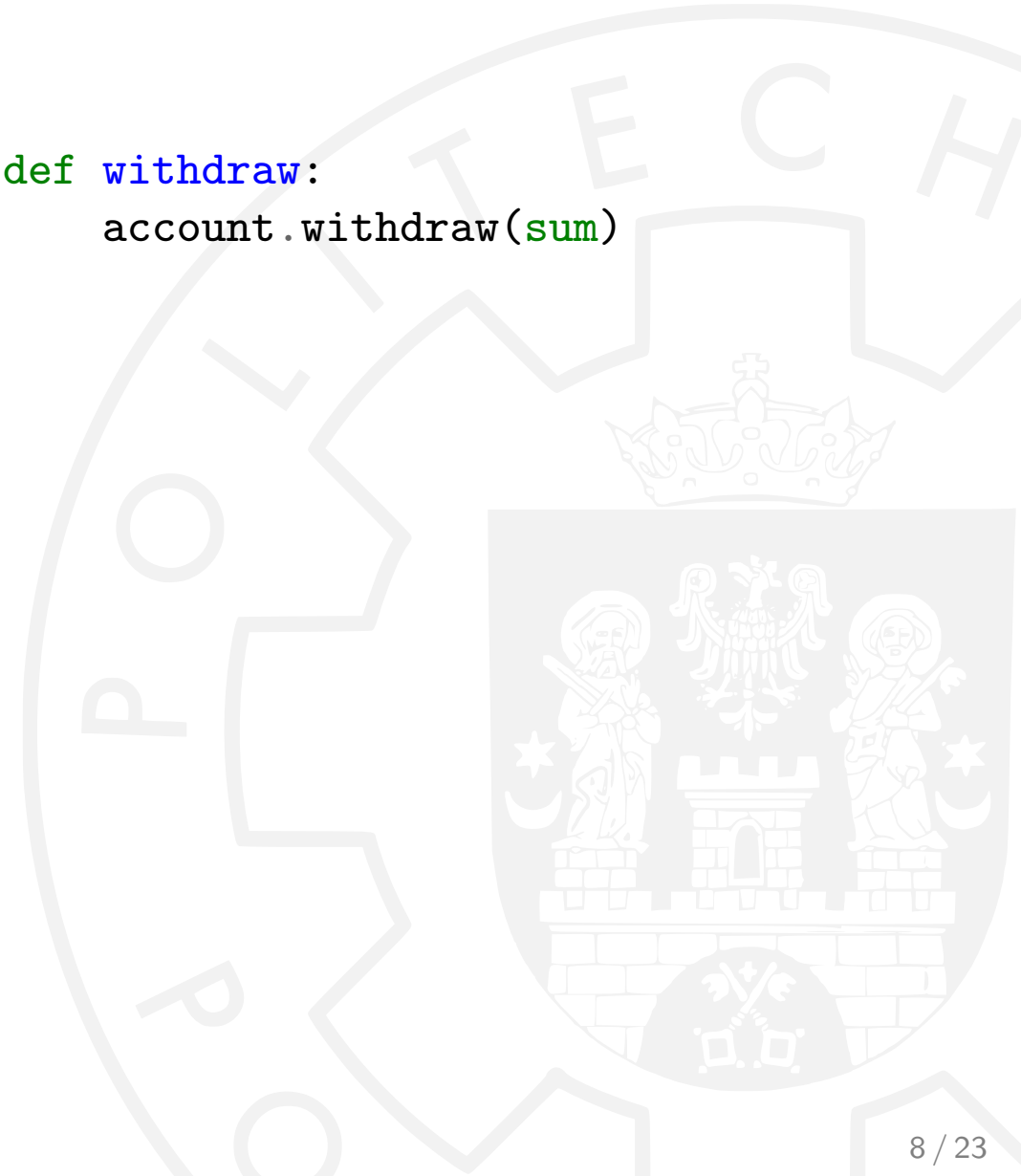


Bank Application

```
def deposit:
    account.deposit(sum)

def balance:
    print account.getBalance()
```

```
def withdraw:
    account.withdraw(sum)
```



Bank Application

```
def deposit:
    account.deposit(sum)

def balance:
    print account.getBalance()
```

```
def withdraw:
    account.withdraw(sum)

def transfer:
    account1.withdraw(sum)
    account2.deposit(sum)
```

Bank Application

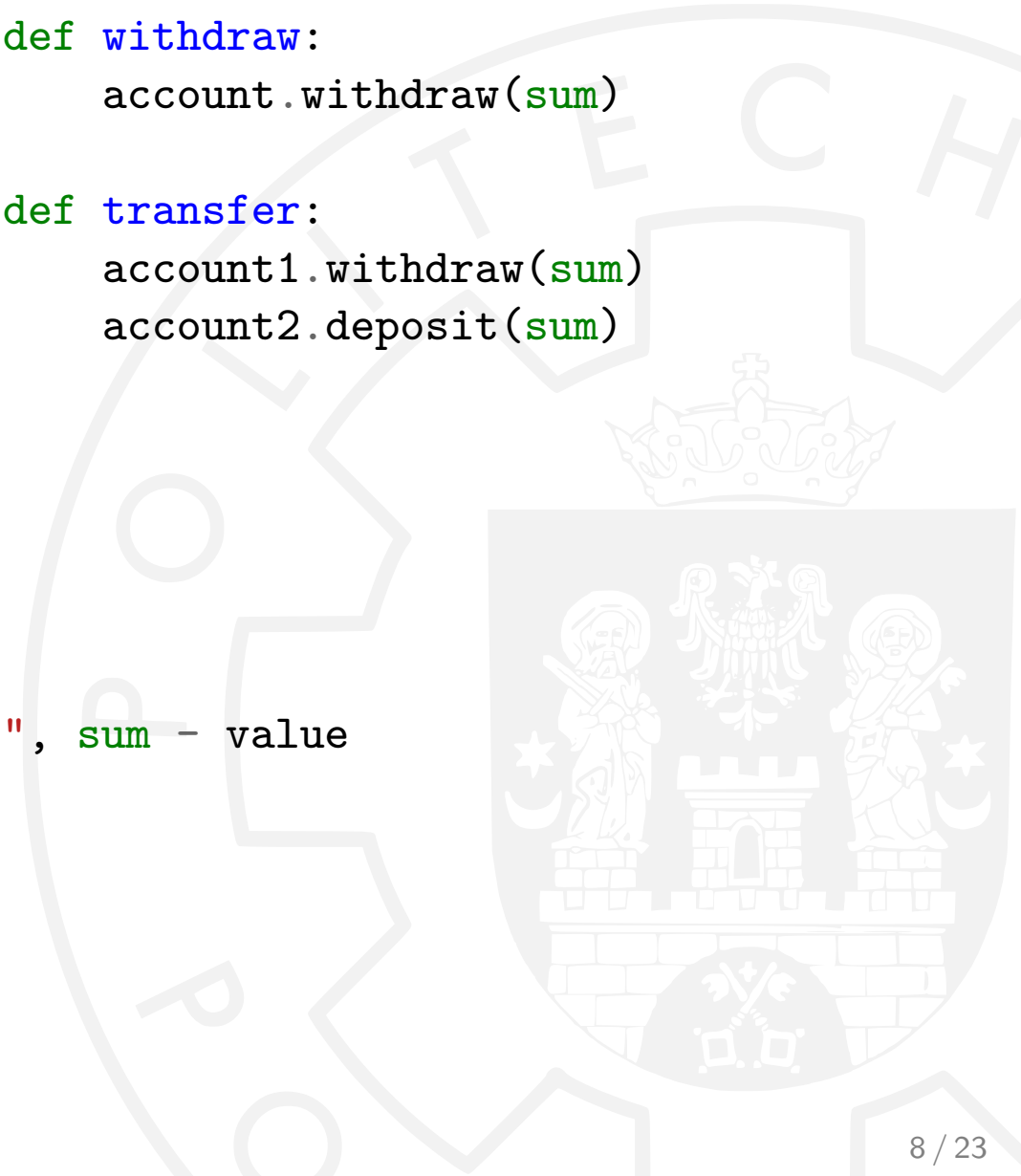
```
def deposit:
    account.deposit(sum)

def balance:
    print account.getBalance()

def audit:
    for a in accounts:
        sum += a.getBalance()
    value = bank.getCapital()
    bank.setCapital(sum)
    print "Accumulated capital", sum - value
```

```
def withdraw:
    account.withdraw(sum)

def transfer:
    account1.withdraw(sum)
    account2.deposit(sum)
```



Bank Application

```
def deposit:
    transaction.start()
    account.deposit(sum)
    transaction.commit()
```

```
def withdraw:
    transaction.start()
    account.withdraw(sum)
    transaction.commit()
```

```
def balance:
    transaction.start()
    print account.getBalance()
    transaction.commit()
```

```
def transfer:
    transaction.start()
    account1.withdraw(sum)
    account2.deposit(sum)
    transaction.commit()
```

```
def audit:
    transaction.start()
    for a in accounts:
        sum += a.getBalance()
    value = bank.getCapital()
    bank.setCapital(sum)
    print "Accumulated capital", sum - value
    transaction.commit()
```

Bank Application

```
def deposit:
    transaction.start()
    account.deposit(sum)
    transaction.commit()
```

```
def withdraw:
    transaction.start()
    account.withdraw(sum)
    transaction.commit()
```

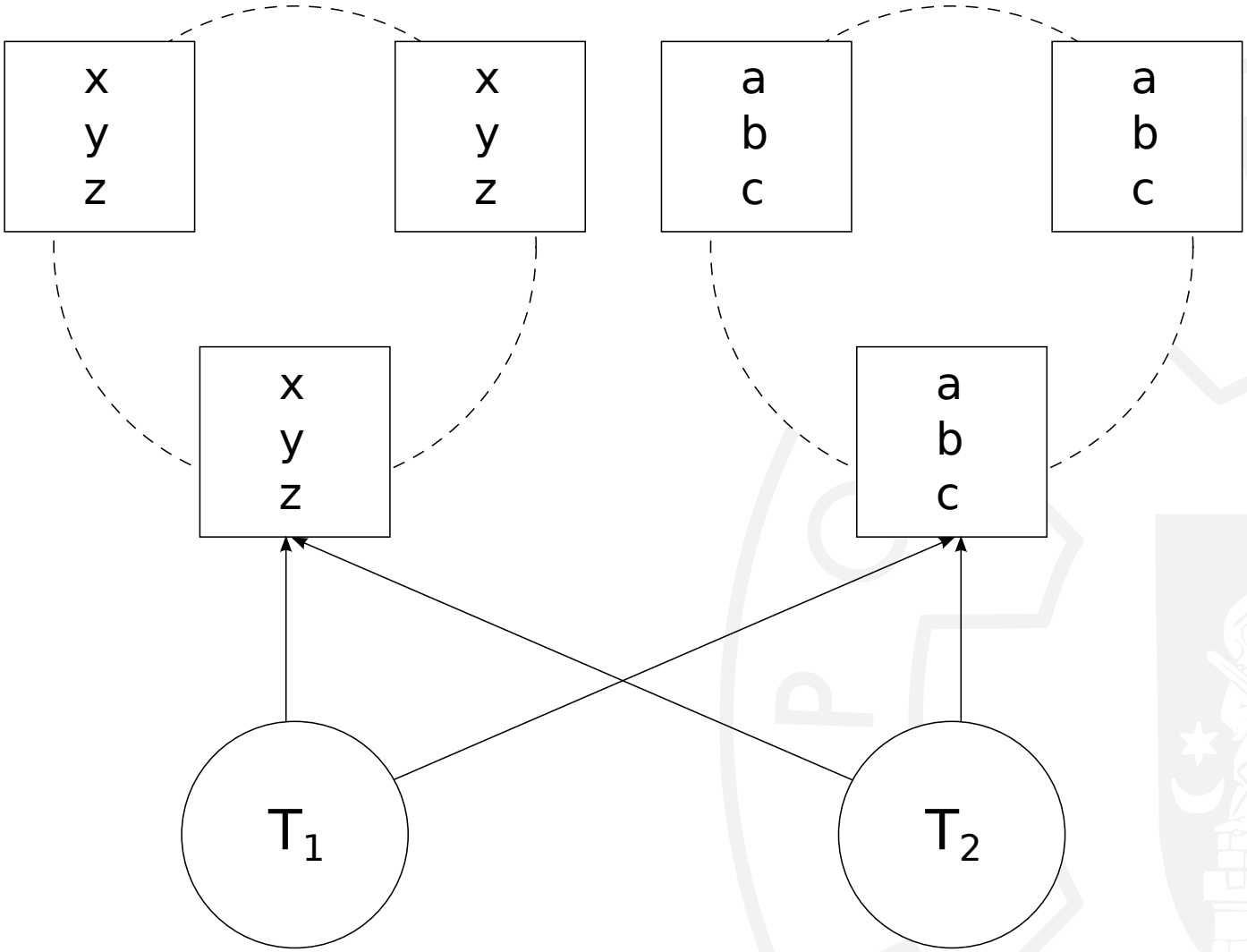
```
def balance:
    transaction.start()
    print account.getBalance()
    transaction.commit()
```

```
def transfer:
    transaction.start()
    account1.withdraw(sum)
    account2.deposit(sum)
    transaction.commit()
```

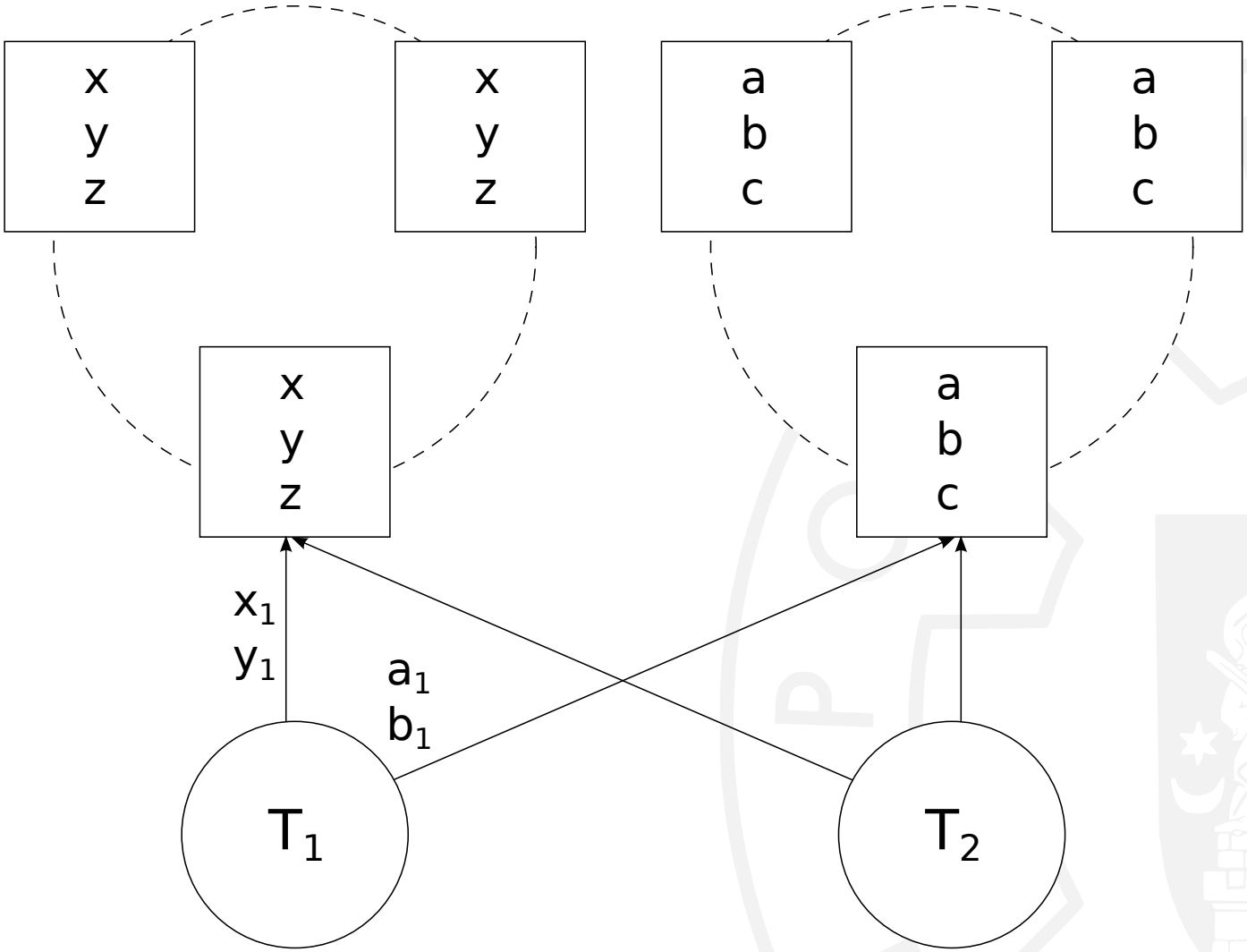
```
def audit:
    transaction.start()
    for a in accounts:
        sum += a.getBalance()
    value = bank.getCapital()
    bank.setCapital(sum)
    print "Accumulated capital", sum - value
    transaction.commit()
```

weaken consistency → improve efficiency

Eventually Consistent Extension



Eventually Consistent Extension

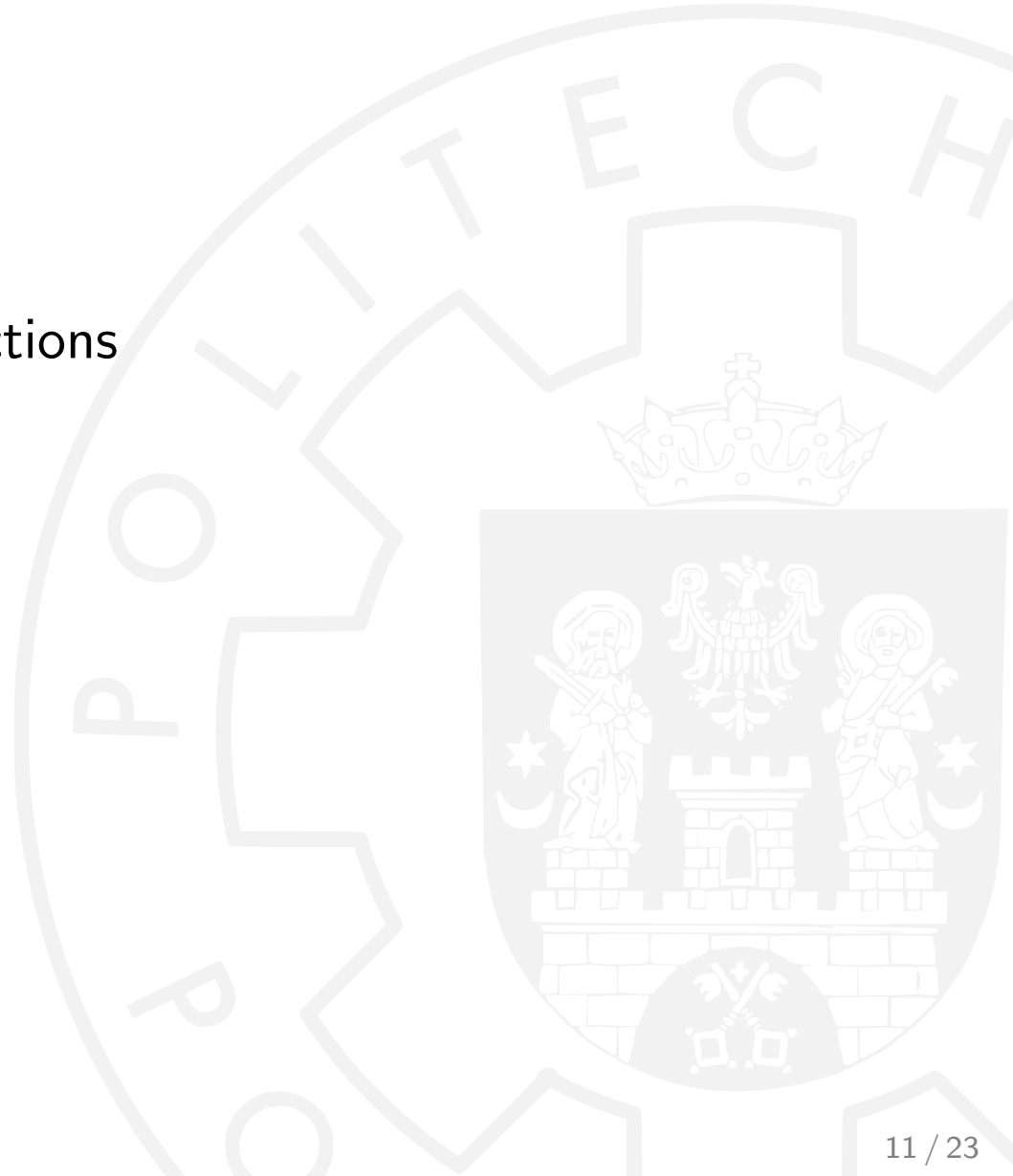


What we require from weak transactions

- do not wait for variables

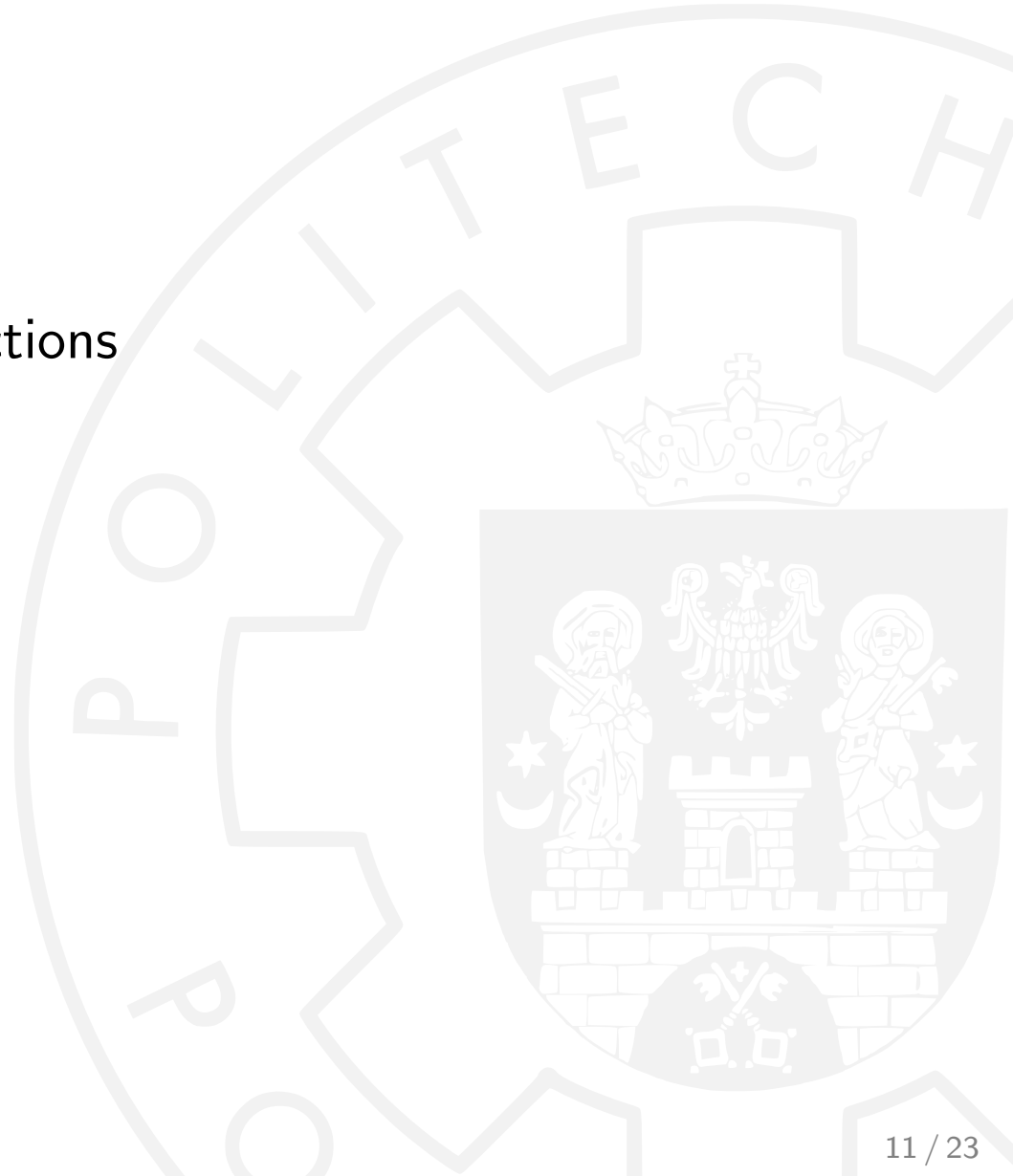
What we require from weak transactions

- do not wait for variables
- do not block other transactions



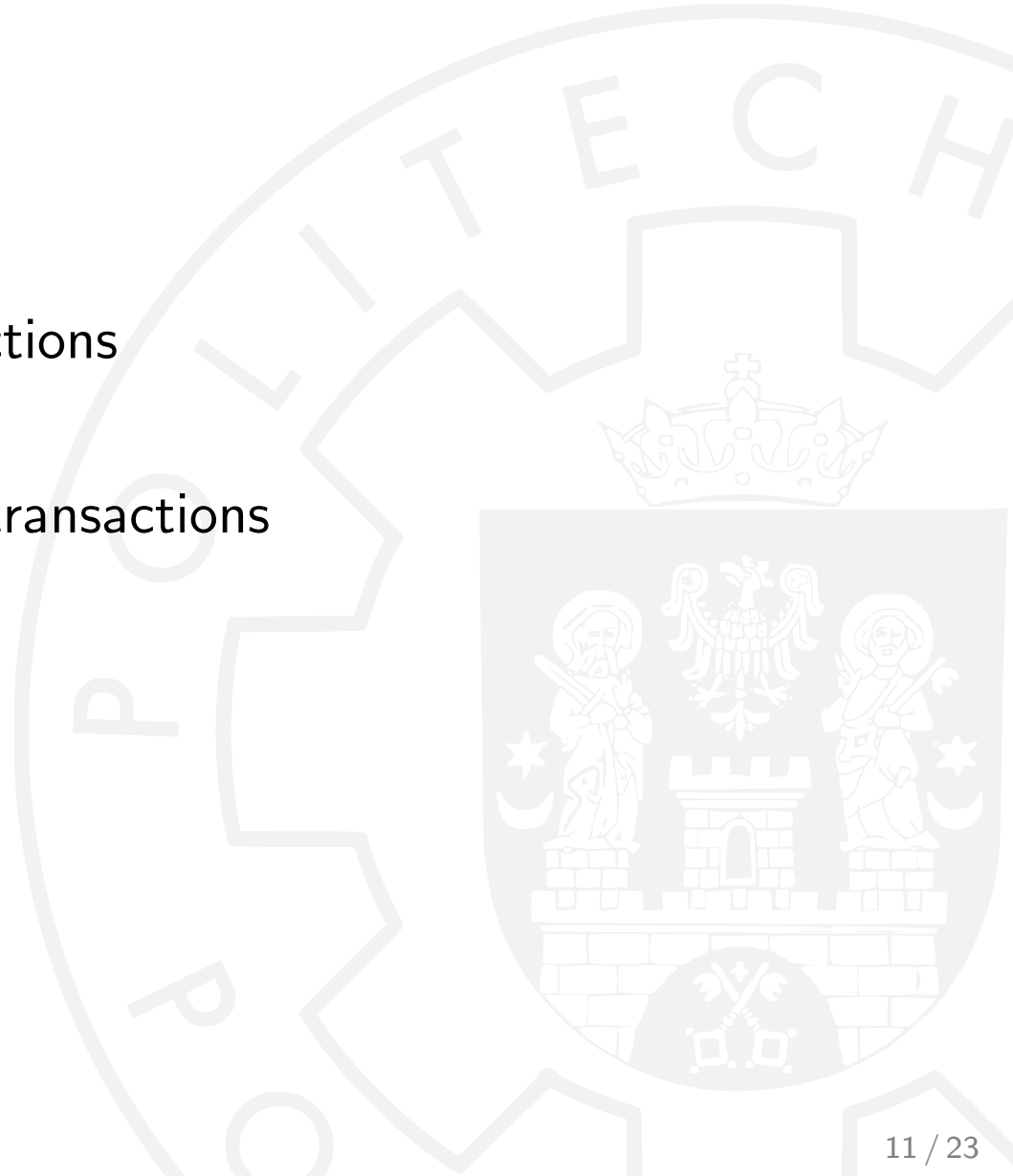
What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency



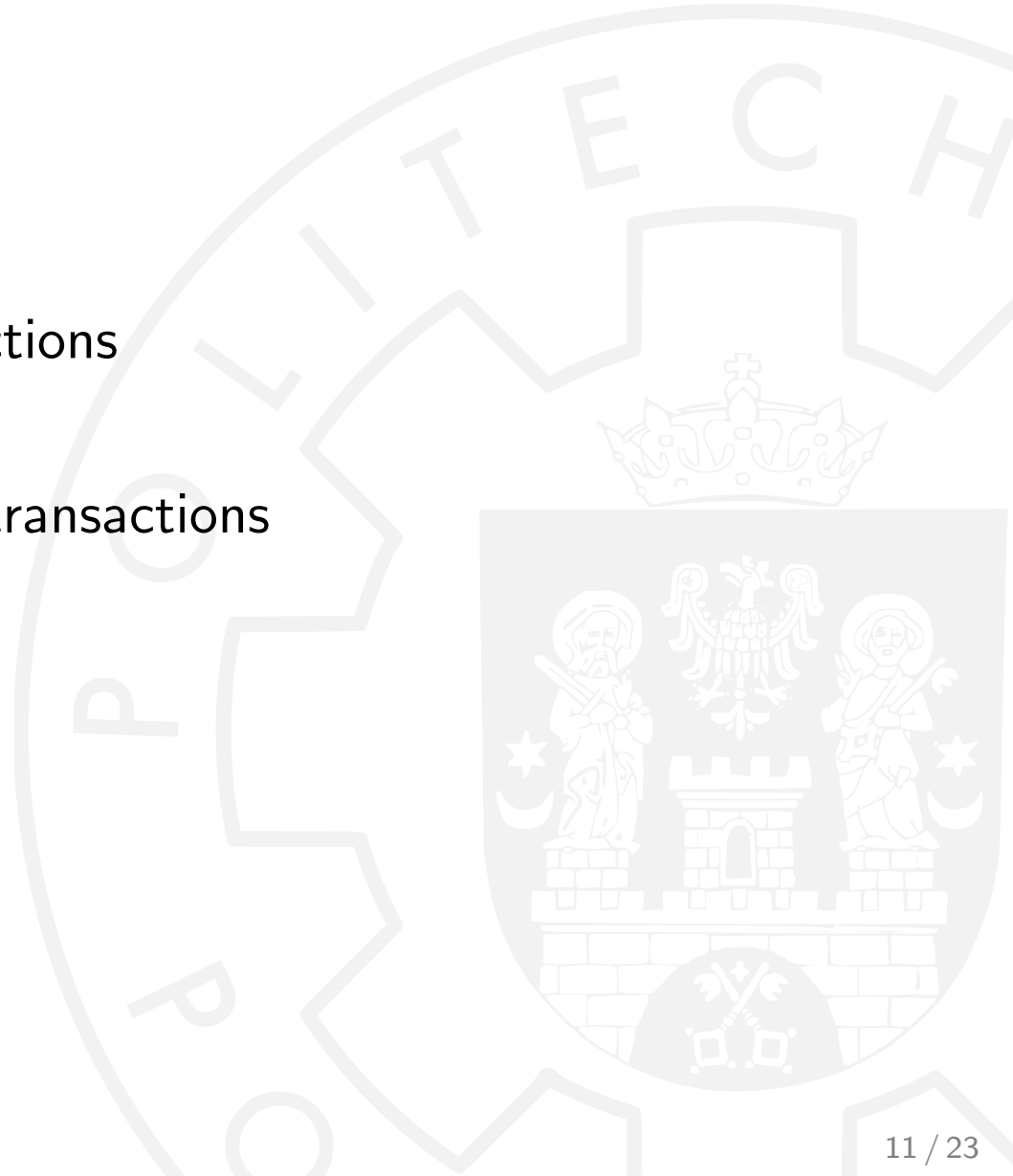
What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency
- do not disturb consistent transactions



What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency
- do not disturb consistent transactions
- converge



Transaction Modes

Transaction T_1

$$T_1 [\![r(x) v_c, w(x) u_c]\!]$$

Transaction Modes

Transaction T_1

$$T_1 [\![r(x) v_c, w(x) u_c]\!]$$

Consitent mode

$$T_1^c [\![r(x) v_c, w(x) u_c]\!]$$

Transaction Modes

Transaction T_1

$$T_1 [\![r(x) v_c, w(x) u_c]\!]$$

Consitent mode

$$T_1^c [\![r(x) v_c, w(x) u_c]\!]$$

Eventually consistent mode

$$T_1^{ec} [\ r(x) v_{ec}, w(x) u_{ec} \ ]$$

Transaction Modes

Transaction T_1

$$T_1 [\![r(x) v_c, w(x) u_c]\!]$$

Consitent mode

$$T_1^c [\![r(x) v_c, w(x) u_c]\!]$$

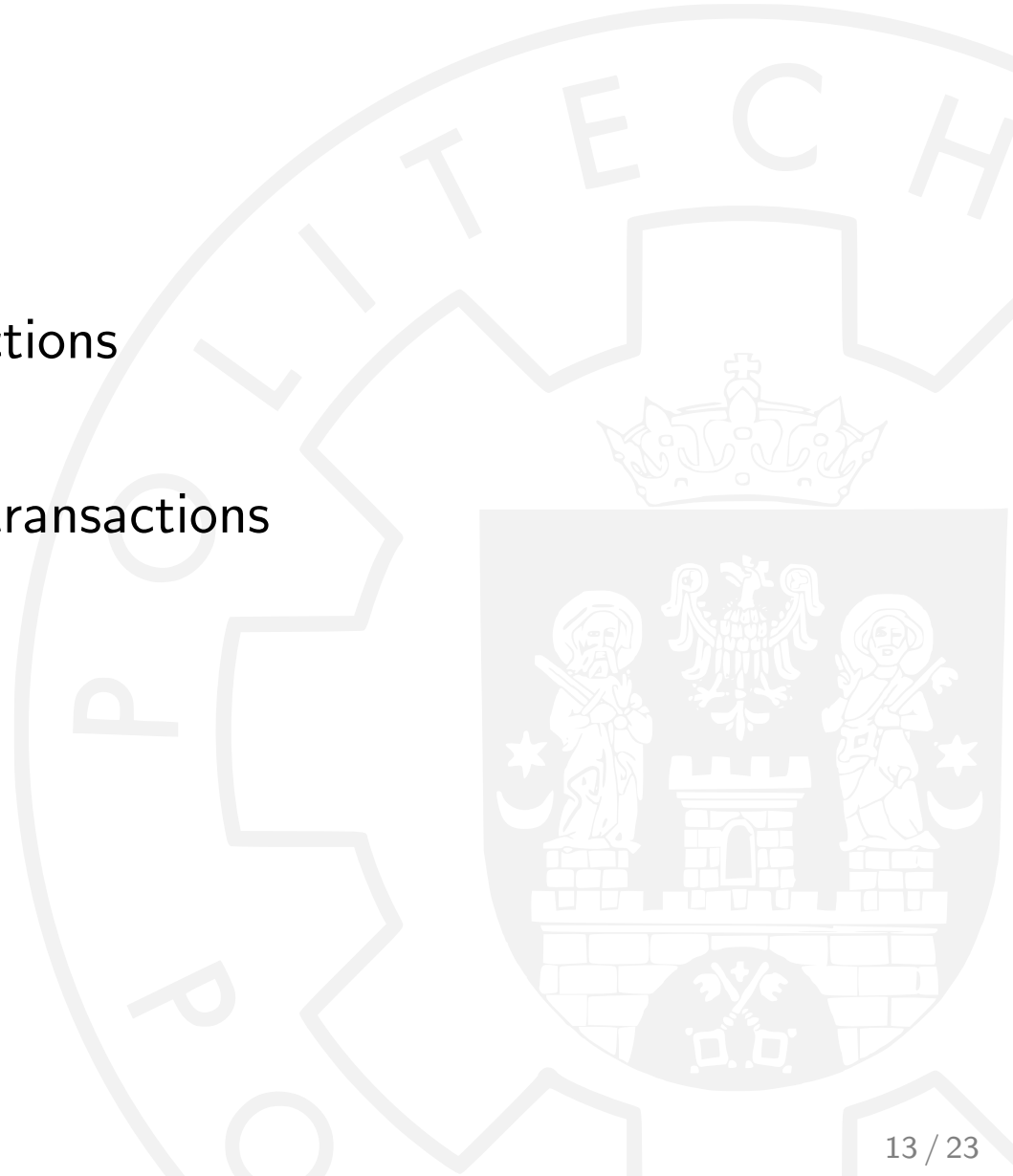
Eventually consistent mode

$$T_1^{ec} [\ r(x) v_{ec}, w(x) u_{ec} \ ]$$

Execute consistent and inconsistent modes simultaneously

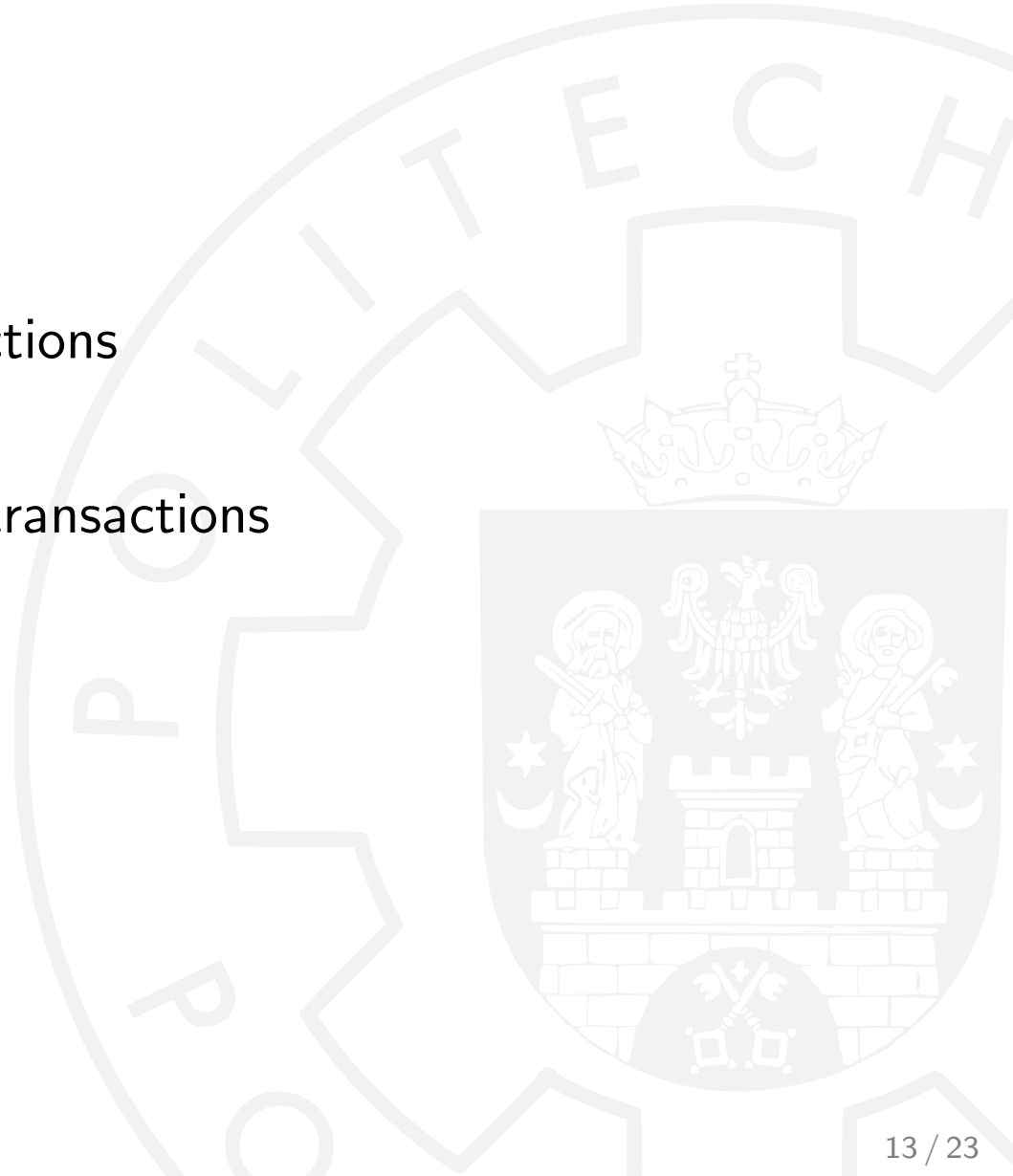
What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency
- do not disturb consistent transactions
- converge



What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency
- do not disturb consistent transactions
- converge ✓



Variable Modification Versions

$$\{x = 1, y = 1\} \quad \begin{array}{ll} T_1 & [\![\, r(x)1, w(x)2, r(y)1, w(y)2 \,]\!] \\ \mid T_2 & [\![\, \searrow r(x)2, w(x)3 \,]\!] \end{array} \quad \{x = 3, y = 2\}$$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\qquad\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

Snapshot Read Consitency

$$T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!]$$
$$| \; T_3 \; [\![\qquad \searrow r(\overset{2}{x})3, w(\overset{3}{x})4, r(\overset{2}{y})3, w(\overset{3}{y})4 \;]\!]$$
$$| \; T_4 \; [\![\qquad \searrow r(\overset{3}{x})4, w(\overset{4}{x})5 \;]\!]$$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

Snapshot Read Consitency

$$T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!]$$
$$| \; T_3 \; [\![\qquad \searrow r(\overset{2}{x})3, w(\overset{3}{x})4, r(\overset{2}{y})3, w(\overset{3}{y})4 \;]\!]$$
$$| \; T_4 \; [\![\qquad \searrow r(\overset{3}{x})4, w(\overset{4}{x})5 \;]\!]$$

$\{\overset{1}{x}, \overset{2}{y}\}$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

Snapshot Read Consitency

$$T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!]$$
$$| \; T_3 \; [\![\qquad \searrow r(\overset{2}{x})3, w(\overset{3}{x})4, r(\overset{2}{y})3, w(\overset{3}{y})4 \;]\!]$$
$$| \; T_4 \; [\![\qquad \searrow r(\overset{3}{x})4, w(\overset{4}{x})5 \;]\!]$$

$\{\overset{1}{x}, \overset{2}{y}\}, \{\overset{2}{x}, \overset{2}{y}\}$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

Snapshot Read Consitency

$$T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!]$$
$$| \; T_3 \; [\![\qquad \searrow r(\overset{2}{x})3, w(\overset{3}{x})4, r(\overset{2}{y})3, w(\overset{3}{y})4 \;]\!]$$
$$| \; T_4 \; [\![\qquad \searrow r(\overset{3}{x})4, w(\overset{4}{x})5 \;]\!]$$

$\{\overset{1}{x}, \overset{2}{y}\}$, $\{\overset{2}{x}, \overset{2}{y}\}$, $\{\overset{3}{x}, \overset{3}{y}\}$, $\{\overset{4}{x}, \overset{3}{y}\}$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

Snapshot Read Consitency

$$T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3 \;]\!]$$
$$| \; T_2 \; [\![\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!]$$
$$| \; T_3 \; [\![\qquad \searrow r(\overset{2}{x})3, w(\overset{3}{x})4, r(\overset{2}{y})3, w(\overset{3}{y})4 \;]\!]$$
$$| \; T_4 \; [\![\qquad \searrow r(\overset{3}{x})4, w(\overset{4}{x})5 \;]\!]$$

$\{\overset{1}{x}, \overset{2}{y}\}$, $\{\overset{2}{x}, \overset{2}{y}\}$, $\{\overset{3}{x}, \overset{3}{y}\}$, $\{\overset{4}{x}, \overset{3}{y}\}$, $\{\overset{3}{x}, \overset{2}{y}\}$

Variable Modification Versions

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1^{ec} \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$| \; T_2 \; [\![\quad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$

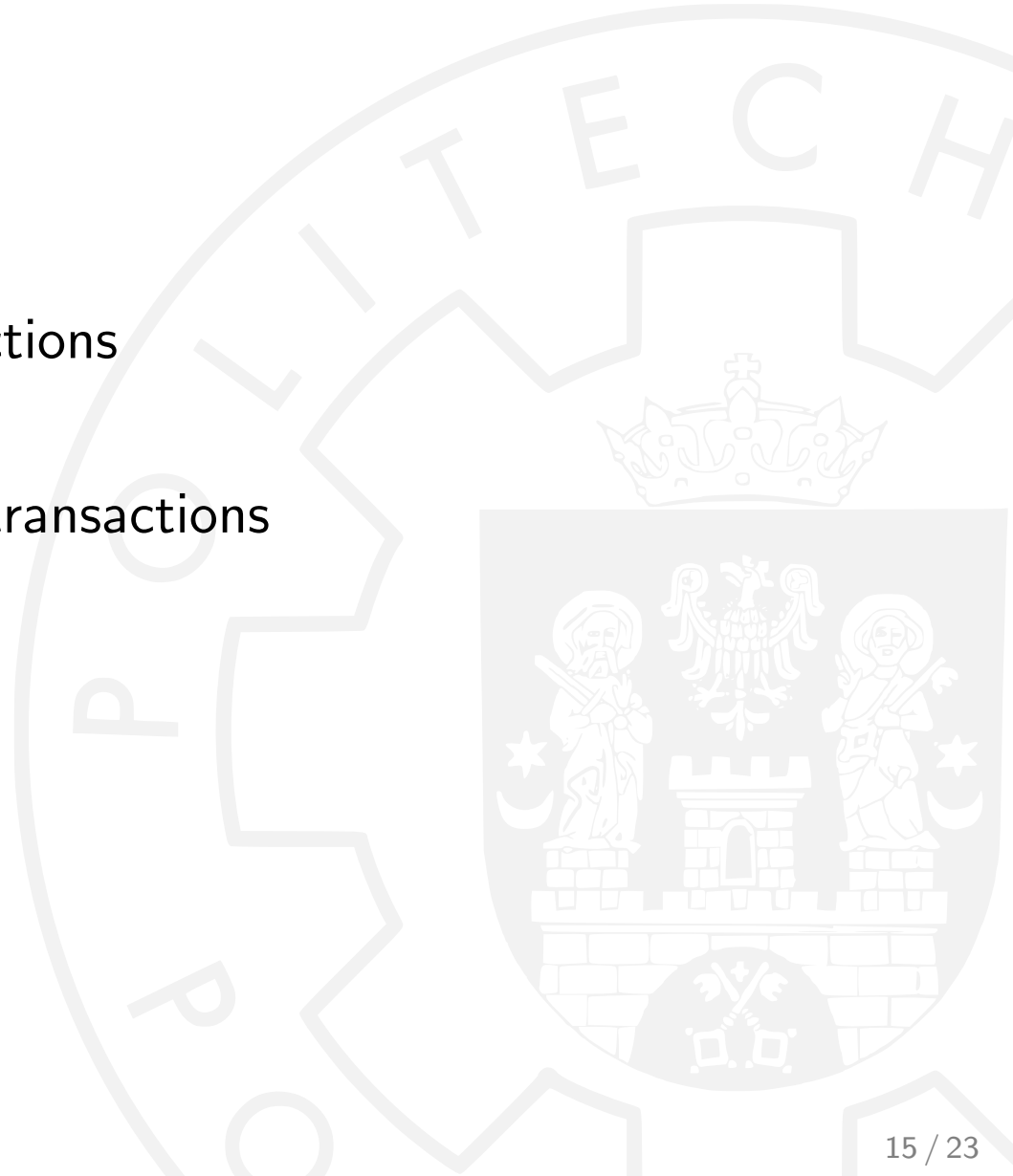
Snapshot Read Consitency

$$T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3 \;]\!]$$
$$| \; T_2 \; [\![\quad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!]$$
$$| \; T_3 \; [\![\quad \searrow r(\overset{2}{x})3, w(\overset{3}{x})4, r(\overset{2}{y})3, w(\overset{3}{y})4 \;]\!]$$
$$| \; T_4 \; [\![\quad \searrow r(\overset{3}{x})4, w(\overset{4}{x})5 \;]\!]$$

$\{\overset{1}{x}, \overset{2}{y}\}$, $\{\overset{2}{x}, \overset{2}{y}\}$, $\{\overset{3}{x}, \overset{3}{y}\}$, $\{\overset{4}{x}, \overset{3}{y}\}$, $\{\overset{3}{x}, \overset{2}{y}\}$, $\{\overset{*}{x}, \overset{1}{y}\}$

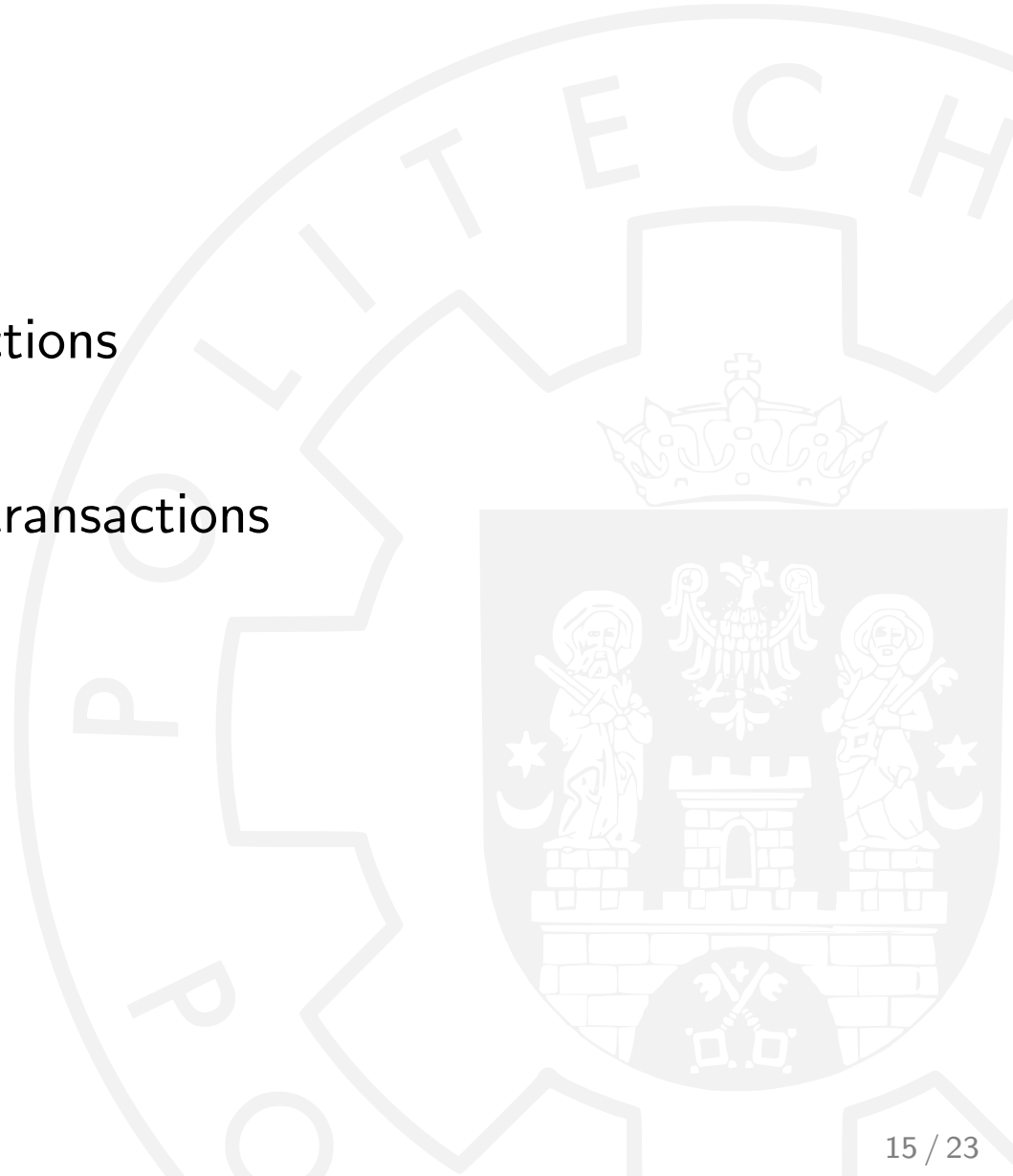
What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency
- do not disturb consistent transactions
- converge ✓



What we require from weak transactions

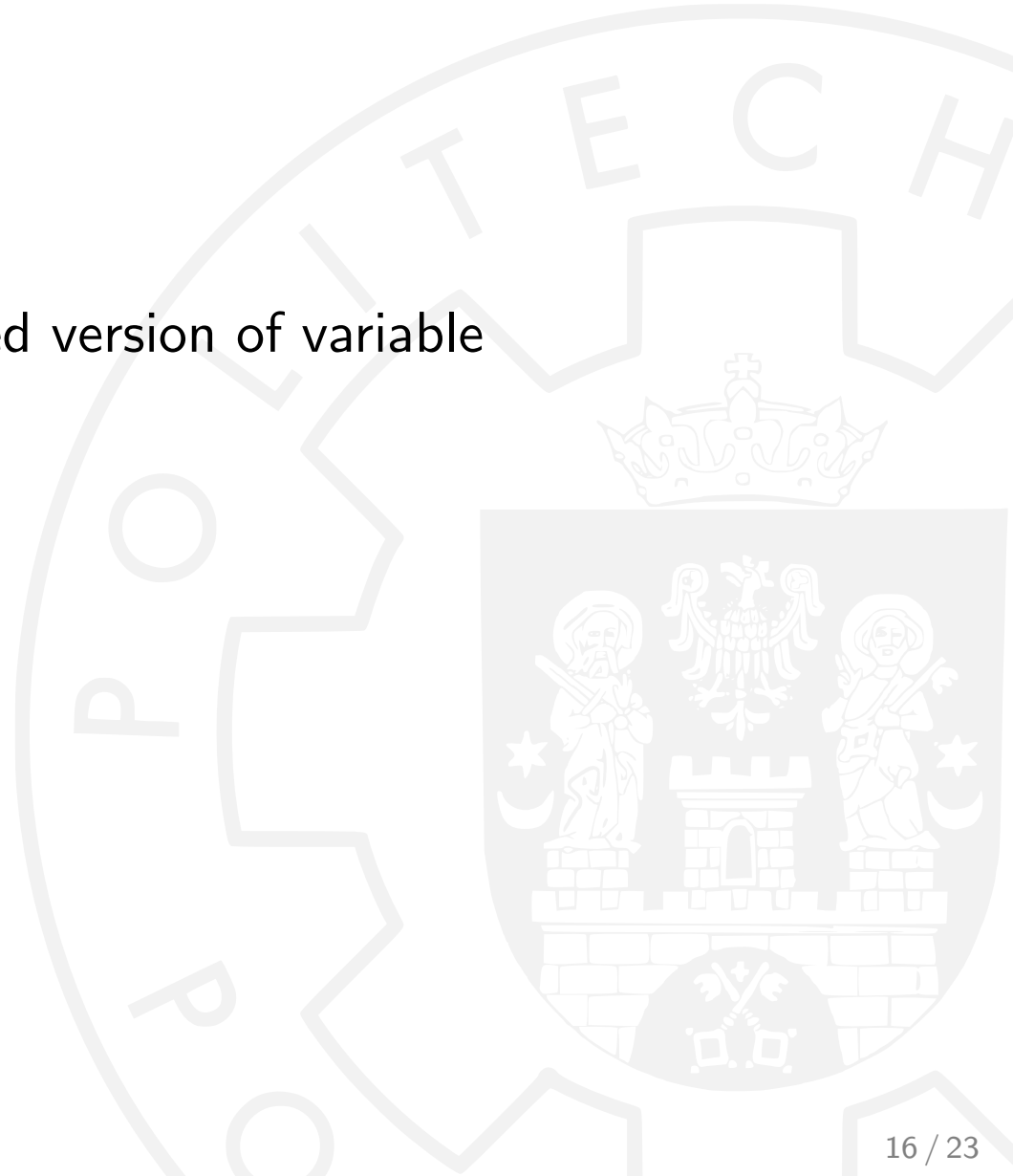
- do not wait for variables
- do not block other transactions
- internal consistency ✓
- do not disturb consistent transactions
- converge ✓



Consistent Snapshot in Practice

Transactions:

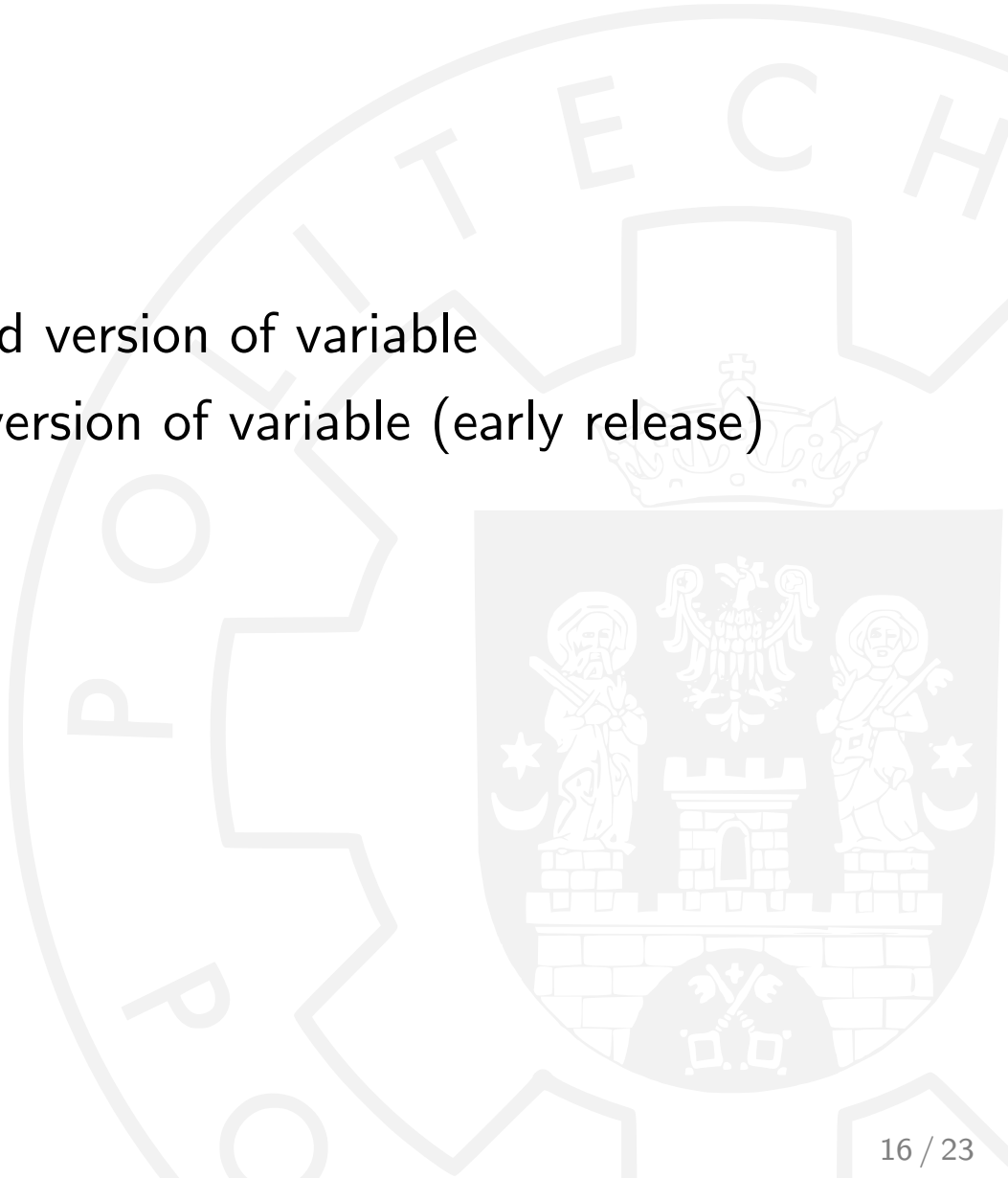
- record the latest committed version of variable



Consistent Snapshot in Practice

Transactions:

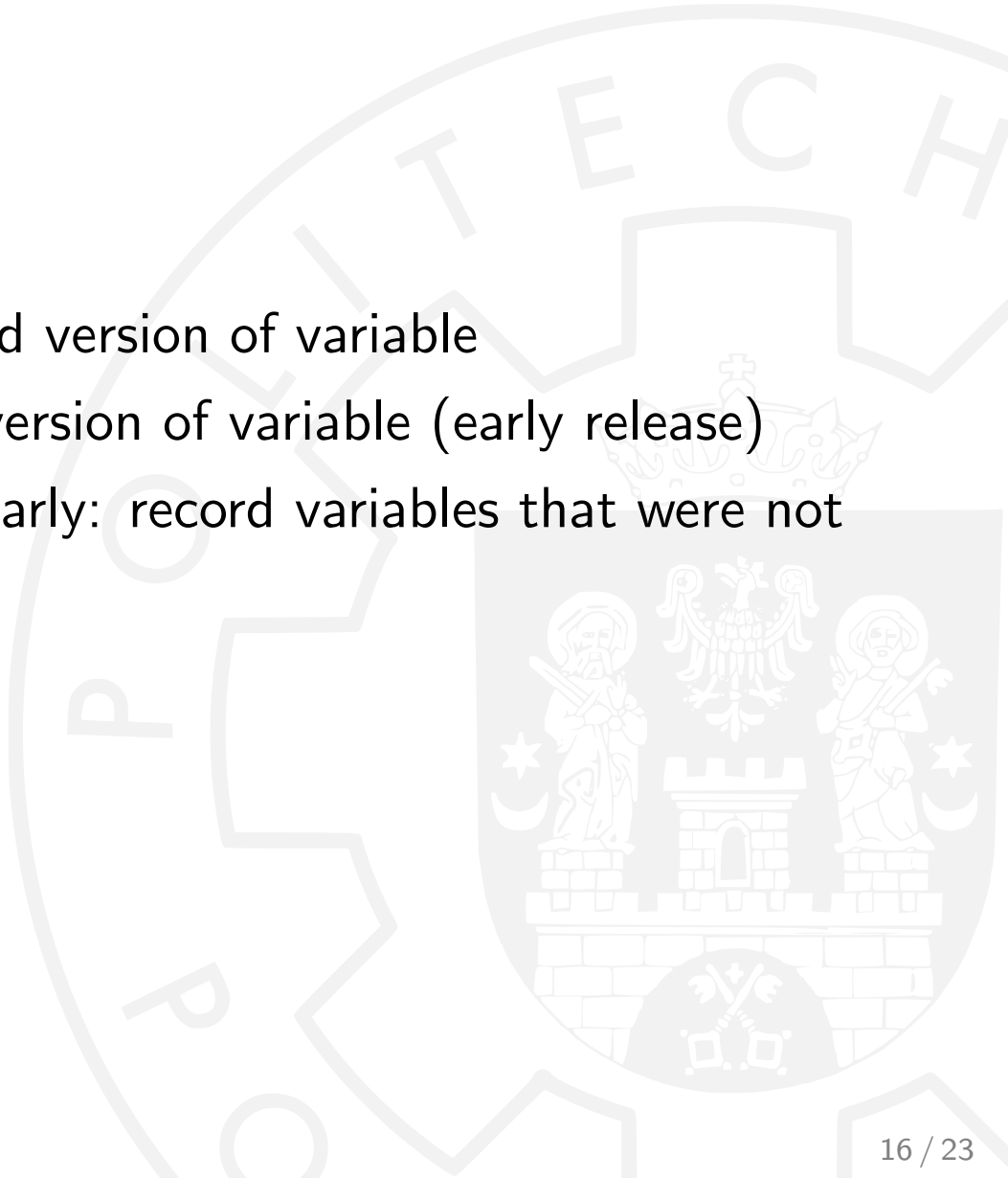
- record the latest committed version of variable
- record the latest released version of variable (early release)



Consistent Snapshot in Practice

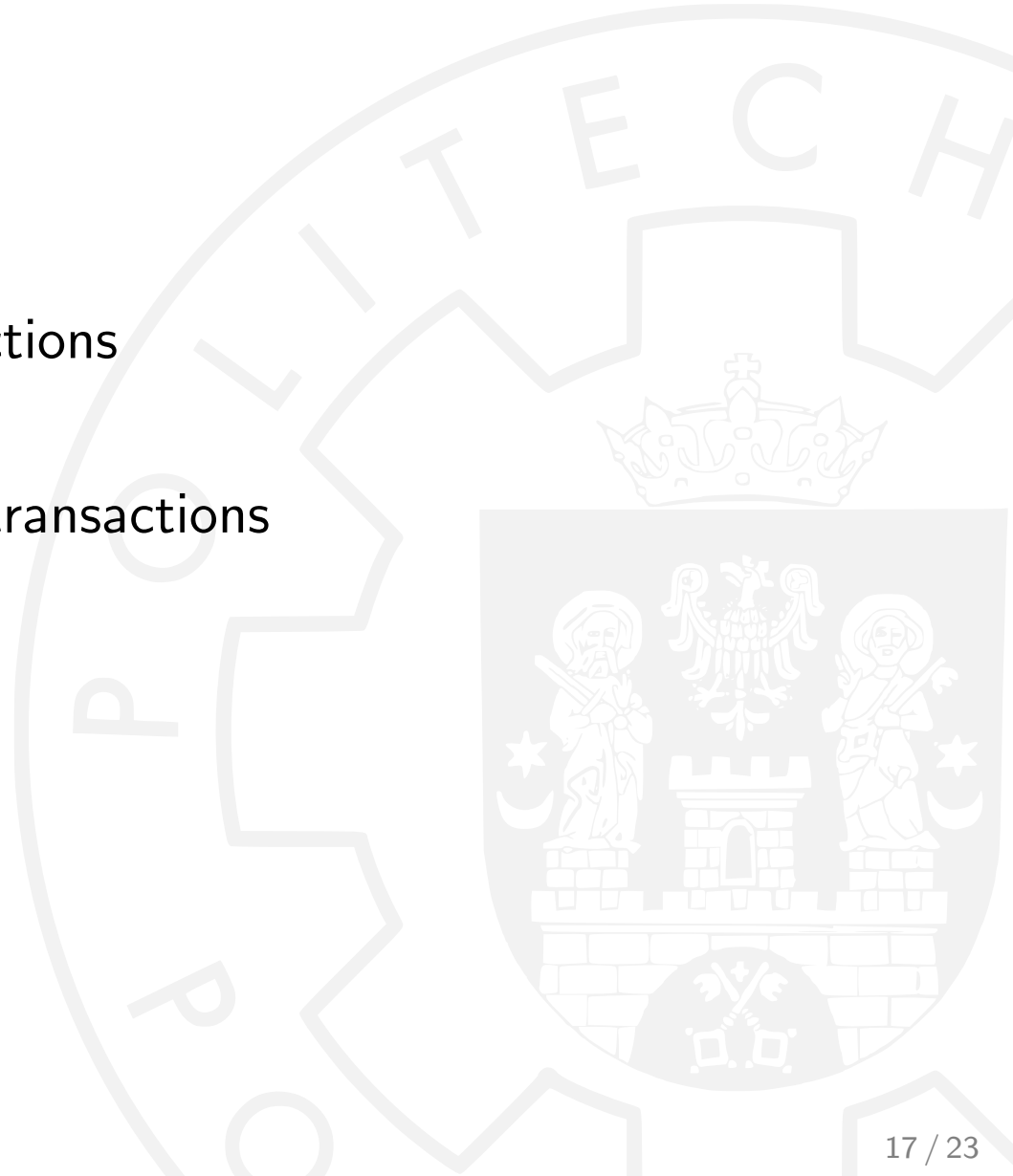
Transactions:

- record the latest committed version of variable
- record the latest released version of variable (early release)
- when releasing a variable early: record variables that were not released early



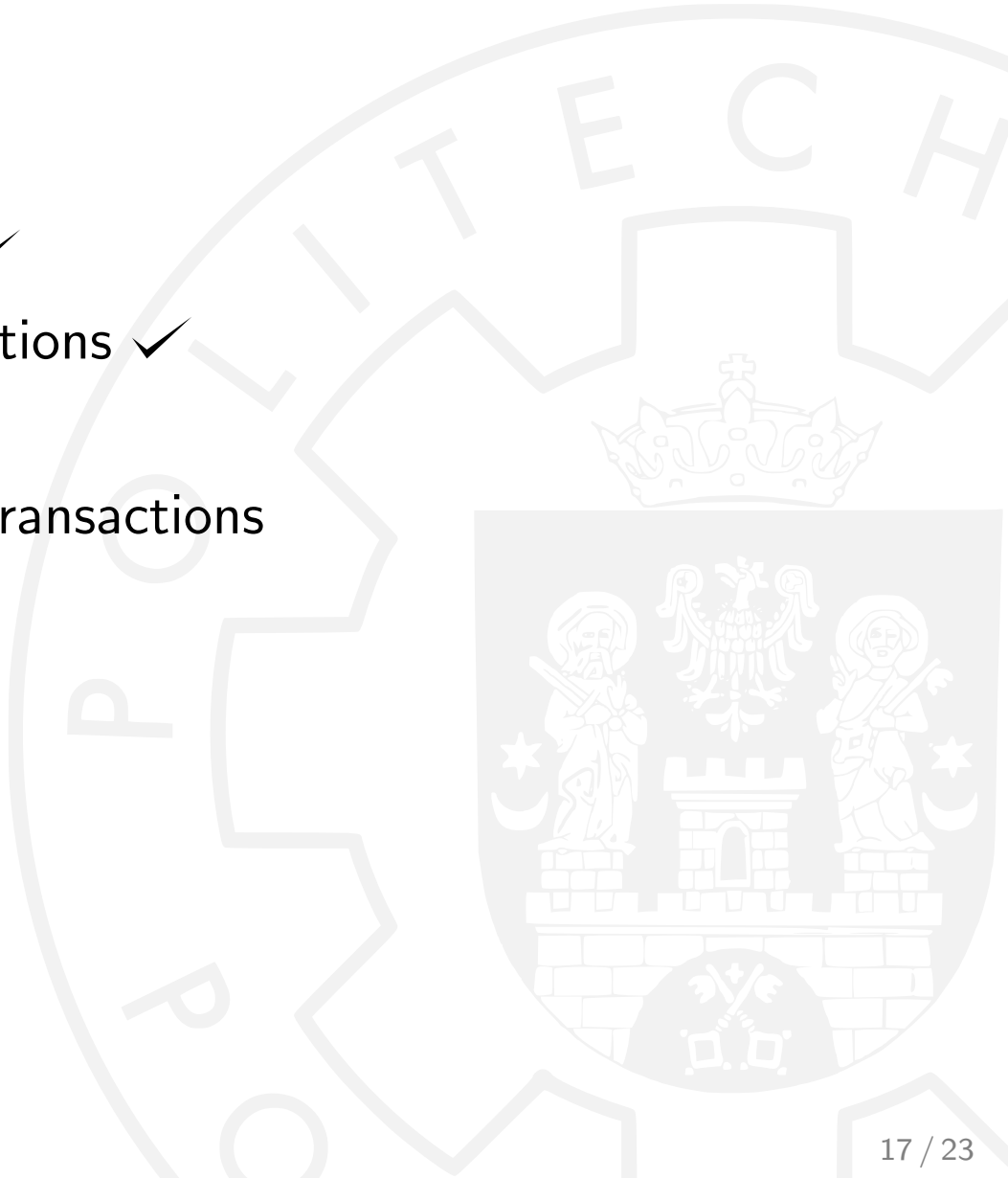
What we require from weak transactions

- do not wait for variables
- do not block other transactions
- internal consistency ✓
- do not disturb consistent transactions
- converge ✓



What we require from weak transactions

- do not wait for variables ✓
- do not block other transactions ✓
- internal consistency ✓
- do not disturb consistent transactions
- converge ✓



Write Bufferring

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\}\ T_1\ [\![\ r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2, w(\overset{2}{y})3\ ]\!]\ \{\overset{1}{x} = 2, \overset{2}{y} = 3\}$$

Write Bufferring

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\}\ T_1\ [\ r(\overset{0}{x})1, w(\underline{x})2, r(\overset{0}{y})1, w(\underline{y})2, w(\underline{y})3\ ]\ \{\overset{0}{x} = 1, \overset{0}{y} = 1\}$$
$$\{\underline{x} = 2, \underline{y} = 3\}$$

Write Bufferring

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\}\ T_1\ [\ r(\overset{0}{x})1, w(\underline{x})2, r(\overset{0}{y})1, w(\underline{y})2, w(\underline{y})3\ ]\ \{\overset{0}{x} = 1, \overset{0}{y} = 1\}$$
$$\{\underline{x} = 2, \underline{y} = 3\}$$

Consistent mode either:

- applies the bufferred writes

Write Bufferring

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\}\ T_1\ [\ r(\overset{0}{x})1, w(\underline{x})2, r(\overset{0}{y})1, w(\underline{y})2, w(\underline{y})3\ ]\ \{\overset{0}{x} = 1, \overset{0}{y} = 1\}$$
$$\{\underline{x} = 2, \underline{y} = 3\}$$

Consistent mode either:

- applies the bufferred writes (if consistency condition allows)

Write Bufferring

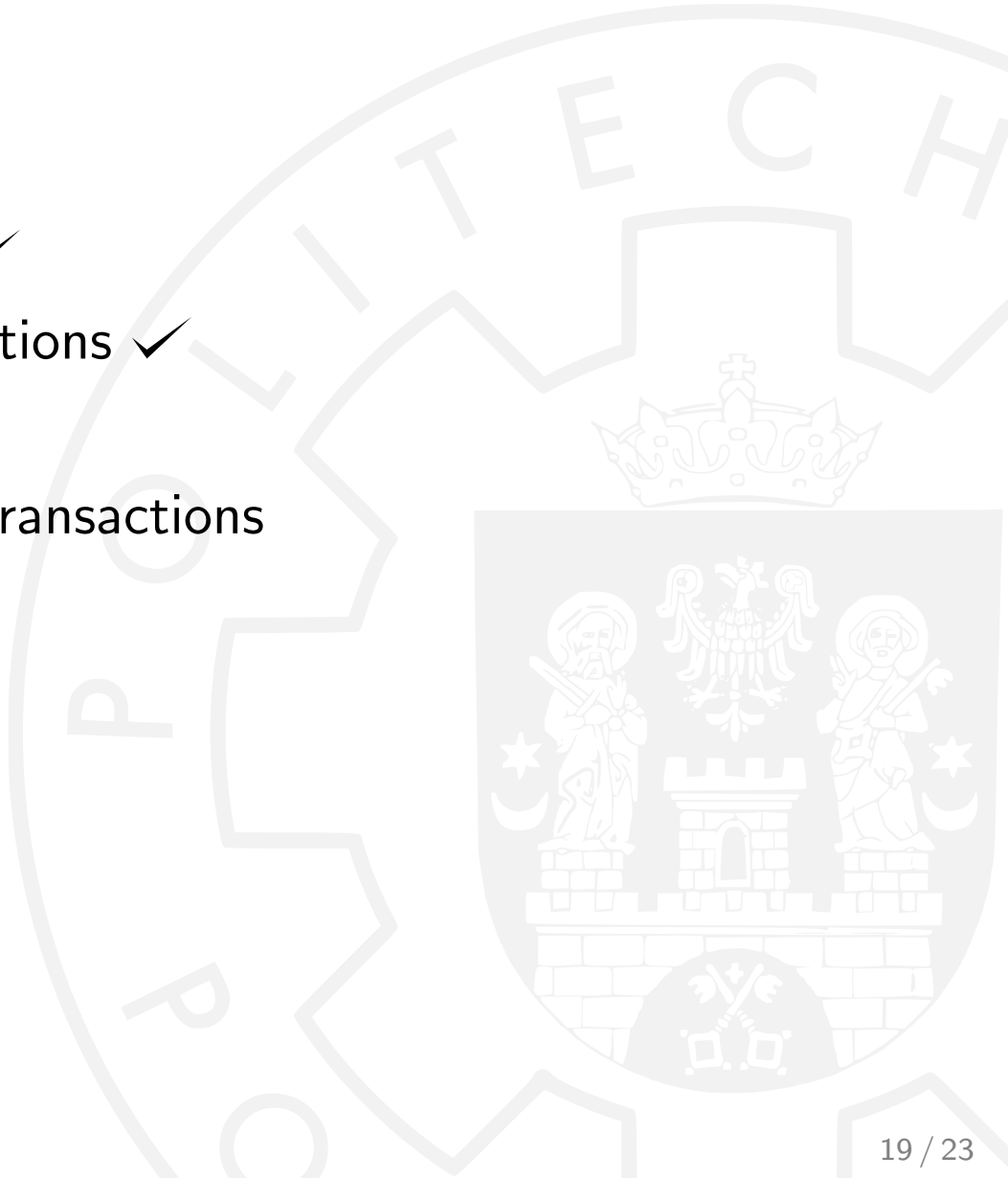
$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\}\ T_1\ [\ r(\overset{0}{x})1, w(\underline{x})2, r(\overset{0}{y})1, w(\underline{y})2, w(\underline{y})3\ ]\ \{\overset{0}{x} = 1, \overset{0}{y} = 1\}$$
$$\{\underline{x} = 2, \underline{y} = 3\}$$

Consistent mode either:

- applies the bufferred writes (if consistency condition allows)
- re-executes from scratch

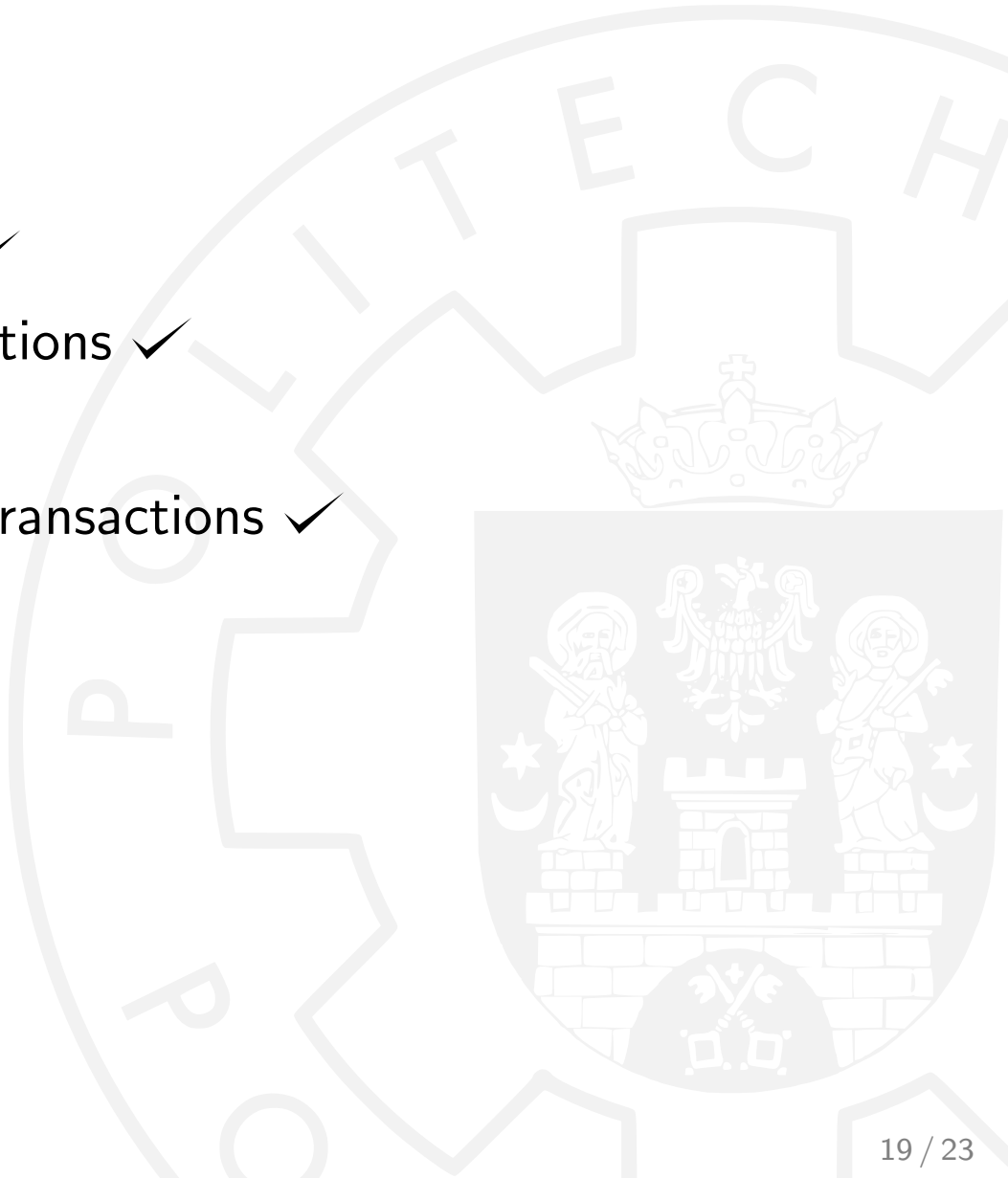
What we require from weak transactions

- do not wait for variables ✓
- do not block other transactions ✓
- internal consistency ✓
- do not disturb consistent transactions
- converge ✓



What we require from weak transactions

- do not wait for variables ✓
- do not block other transactions ✓
- internal consistency ✓
- do not disturb consistent transactions ✓
- converge ✓



Eventually Consistent SVA

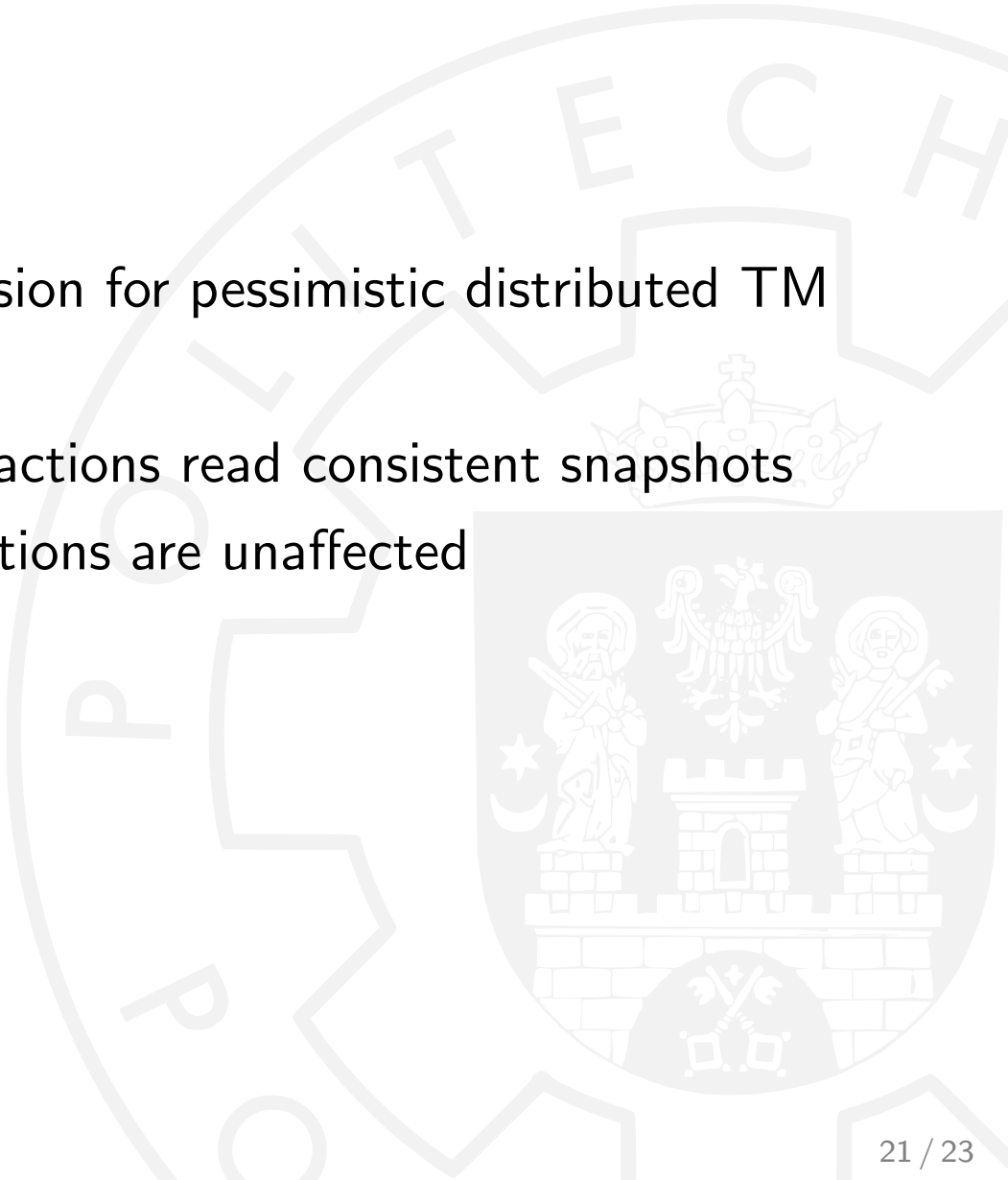
$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$\mid T_2^c \, [\![\qquad\qquad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$
$$\mid T_2^{ec} [\; r(\overset{0}{x})1, w(\underline{x})2 \;]$$

Eventually Consistent SVA

$$\{\overset{0}{x} = 1, \overset{0}{y} = 1\} \quad T_1 \; [\![\; r(\overset{0}{x})1, w(\overset{1}{x})2, r(\overset{0}{y})1, w(\overset{1}{y})2 \;]\!]$$
$$\mid T_2^c \, [\![\quad \searrow r(\overset{1}{x})2, w(\overset{2}{x})3 \;]\!] \quad \{\overset{2}{x} = 3, \overset{1}{y} = 2\}$$
$$\mid T_2^{ec} [\; r(\overset{0}{x})1, w(\underline{x})2 \;] \quad \searrow$$
$$\mid T_3 \quad [\![\; r(\overset{2}{x})3, w(\overset{3}{x})4 \;]\!]$$

Summary

- eventual consistency extension for pessimistic distributed TM
- minimal extra cost
- eventually consistent transactions read consistent snapshots
- strongly consistent transactions are unaffected



Related Papers:

Konrad Siek, Paweł T. Wojciechowski. *Brief Announcement: Towards a Fully-Articulated Pessimistic Distributed Transactional Memory*. In Proceedings of SPAA 2013: the 25th ACM Symposium on Parallelism in Algorithms and Architectures. July 2013.

Paweł T. Wojciechowski, Olivier Rütti and André Schiper. *SAMOA: A Framework for a Synchronisation-Augmented Microprotocol Approach.* In the Proceedings of IPDPS 2004: the 18th IEEE Parallel and Distributed Processing Symposium. April 2004.

?