Runtime Monitoring

of Dynamic Fairness Properties

Thomas A. Henzinger | Mahyar Karrabi | Konstantin Kueffner | Kaushik Mallik

Institute of Science and Technology Austria

Dynamic Fairness.

Is a deployed system fair in the long-run?

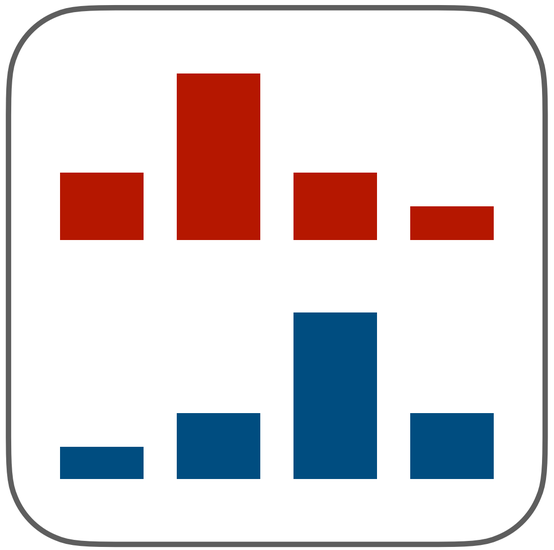
Why?

It was shown that
agents which are fair in a static setting
may hurt protected groups in a dynamic setting.
(D'Amour et al. and Liu et al.)

Liu et al. 2018. Delayed Impact of Fair Machine Learning
D'Amour et al. 2020. Fairness is not static: deeper understanding of long term fairness via simulation studies

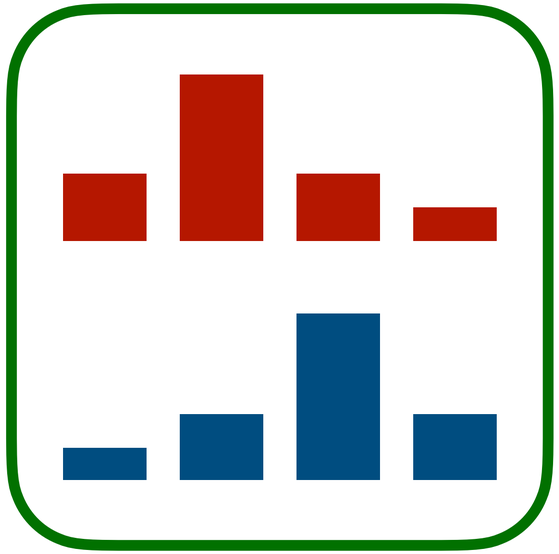
Example.

Dynamic Lending Problem (D'Amour 2020).



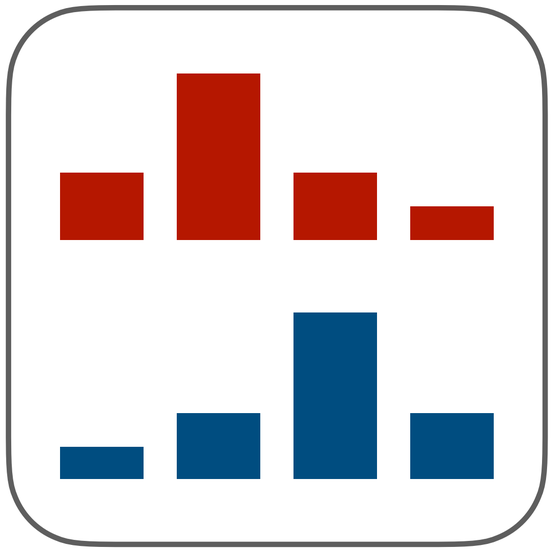
Bank:
g*rant*
or
d*eny*

Customer:
r*epay*
or
d*efault*



Bank:
g*rant*
or
d*eny*

Customer:
r*epay*
or
d*efault*



Bank:

g*rant*

or

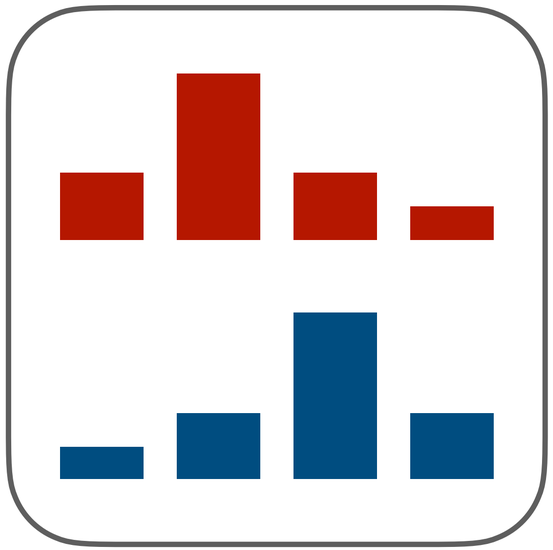
d*eny*

Customer:

r*epay*

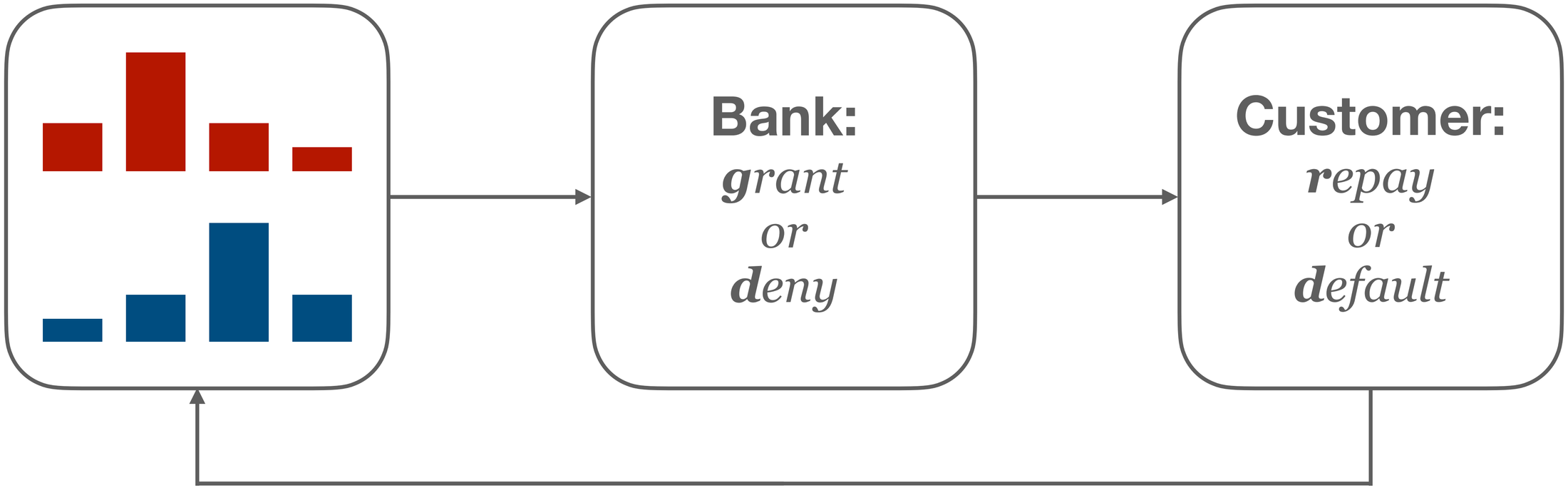
or

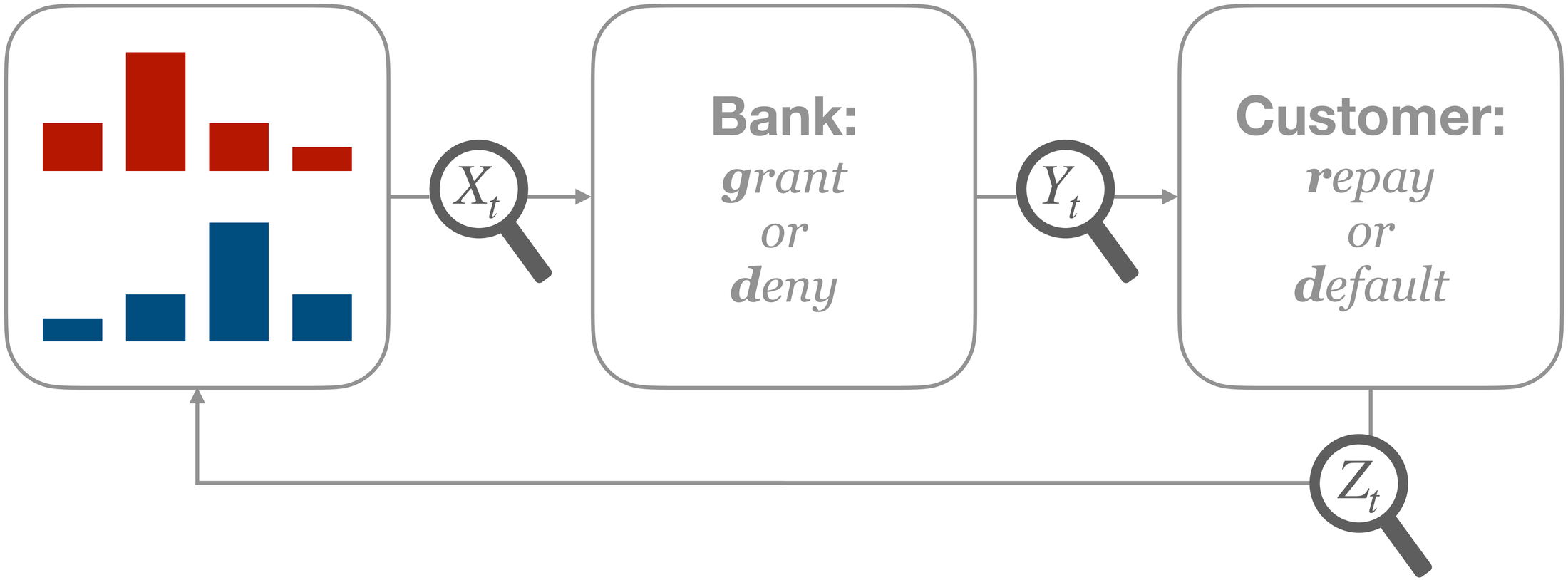
d*efault*

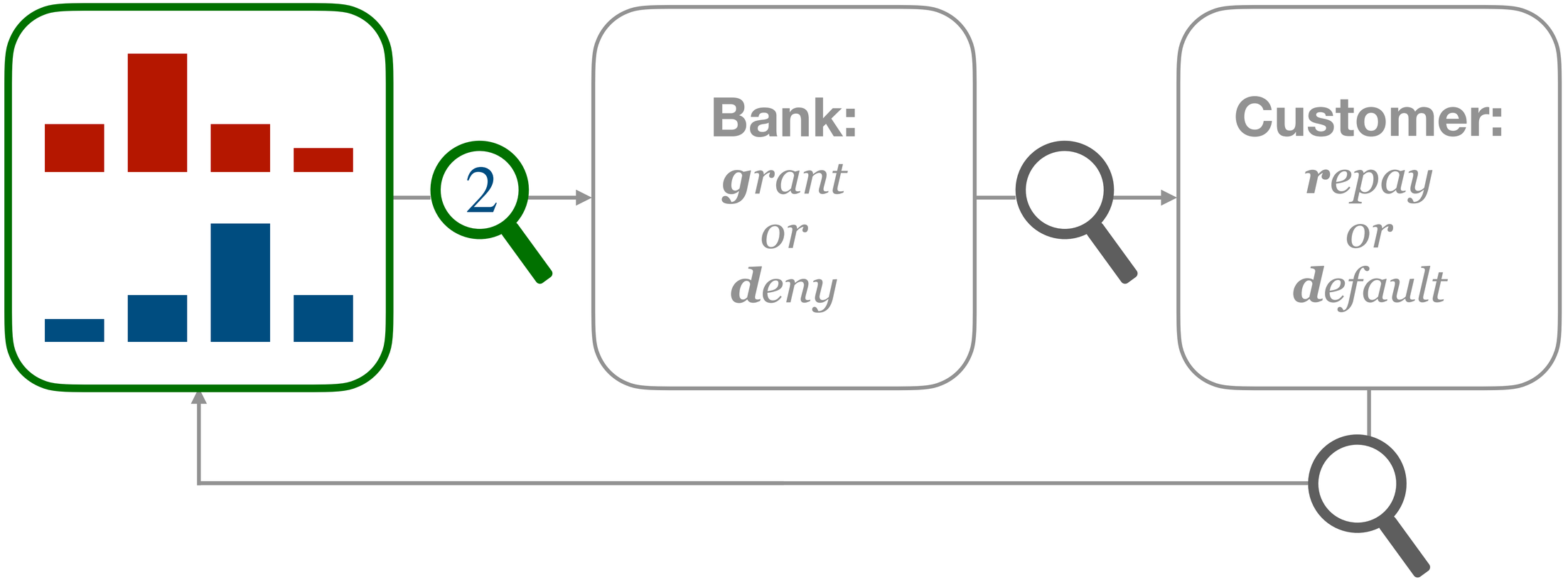


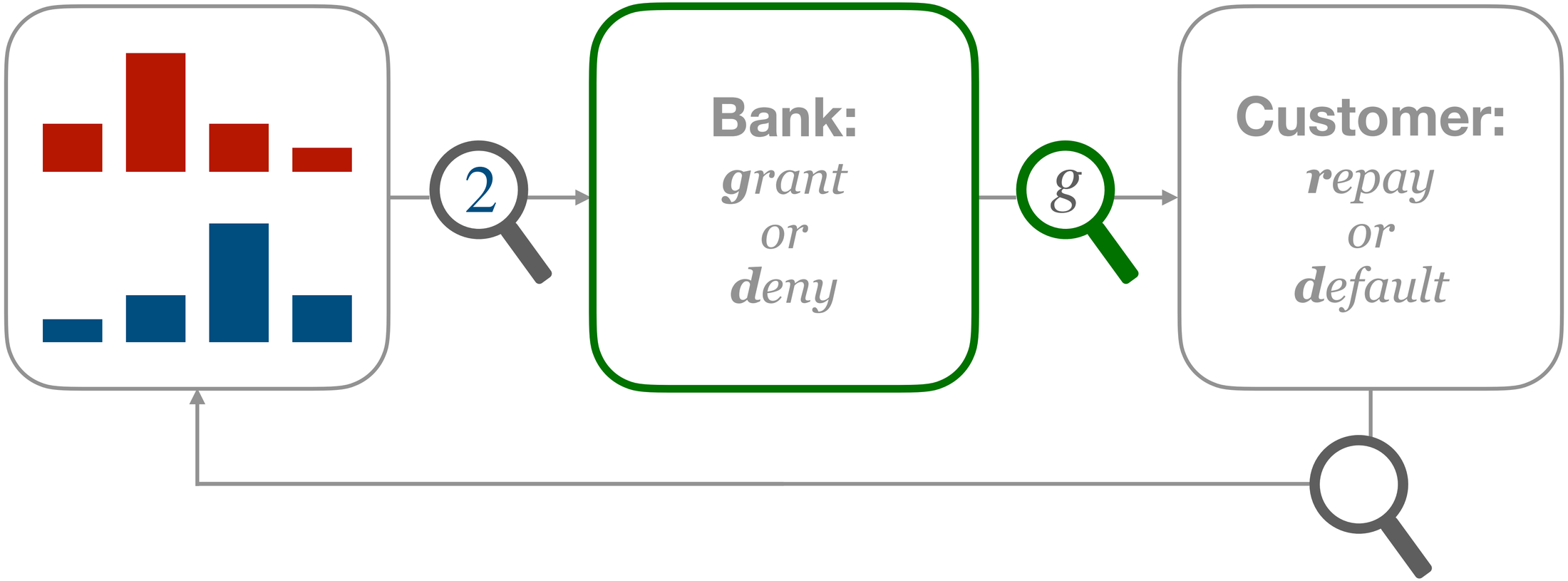
Bank:
g*rant*
or
d*eny*

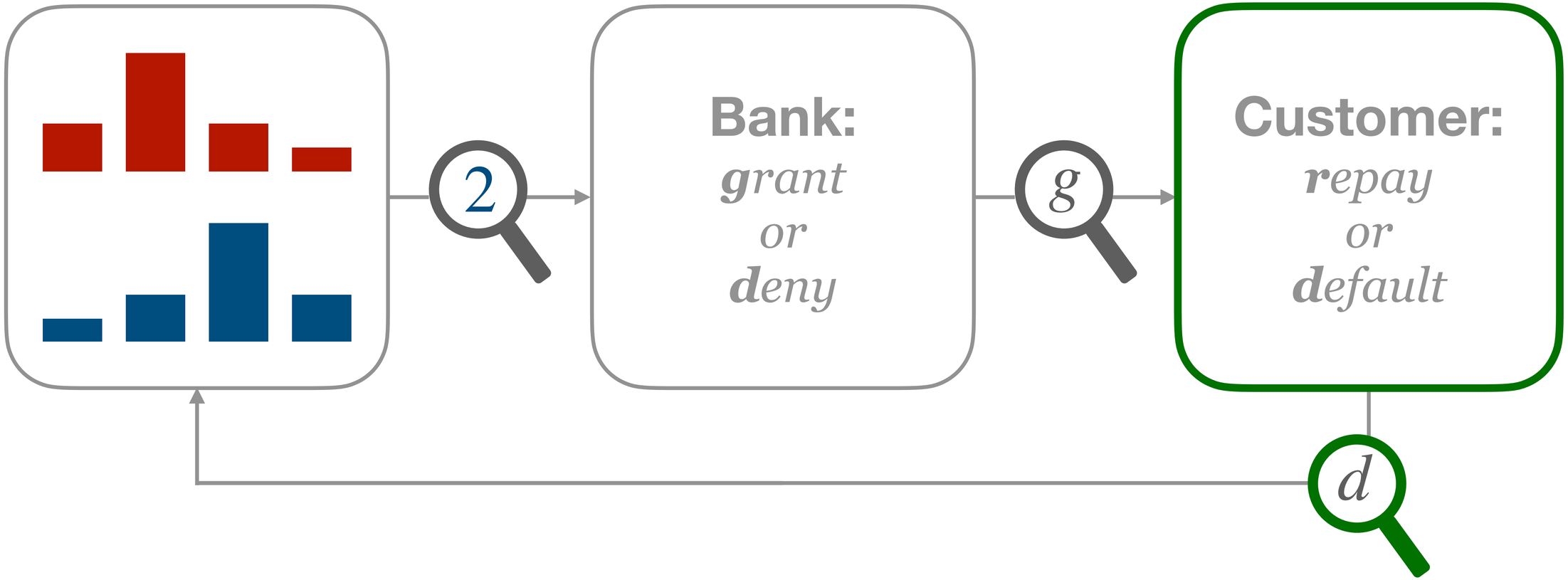
Customer:
r*epay*
or
d*efault*

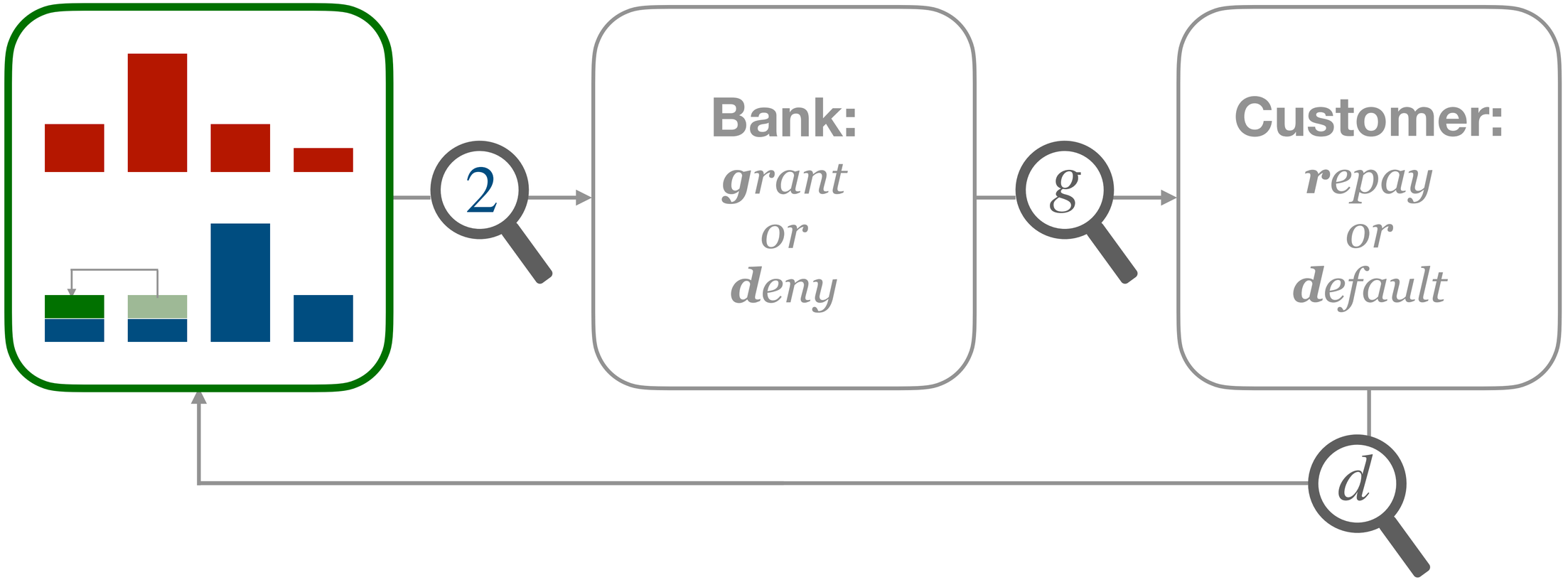

Bank:
grant
or
deny
Customer:
repay
or
default


Bank:
grant
or
deny
Customer:
repay
or
default
X_t
Y_t
Z_t


2
Bank:
grant
or
deny
Customer:
repay
or
default

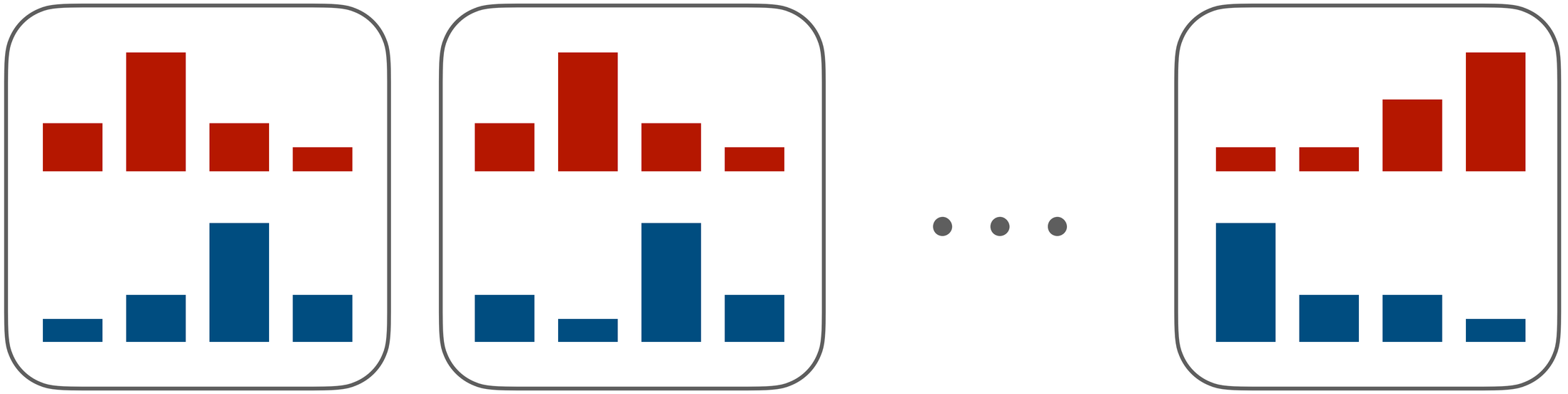

2
Bank:
grant
or
deny
g
Customer:
repay
or
default


2
Bank:
grant
or
deny
g
Customer:
repay
or
default
d



$$\textit{Group Red:} \quad 2.2 \xrightarrow[0]{} 2.2$$

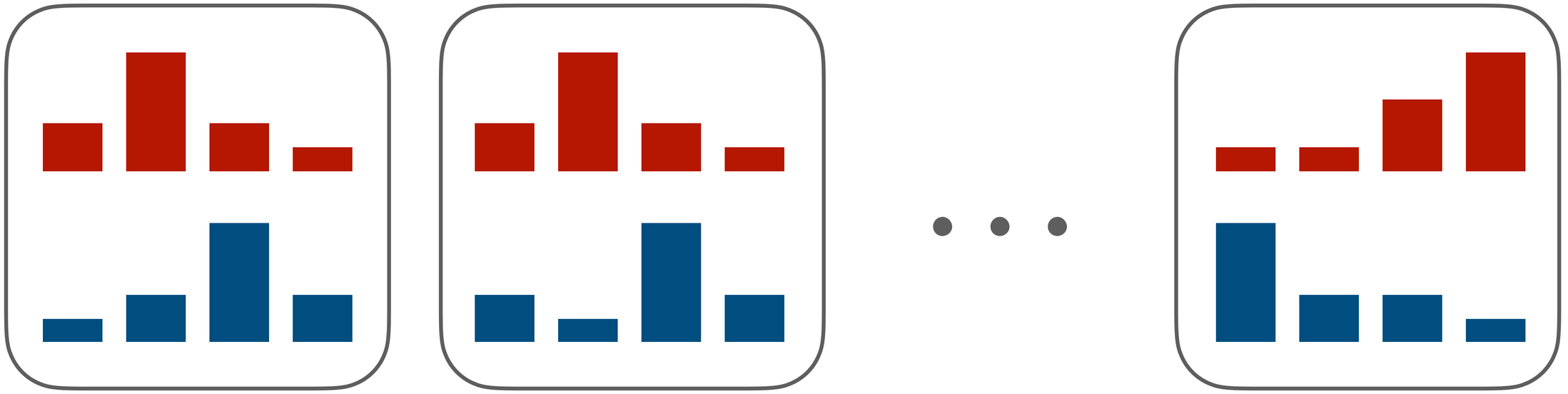
$$\textit{Group Blue:} \quad 2.8 \xrightarrow[\frac{1}{10}]{} 2.7$$



Time 1 *Time 2* *Time T*

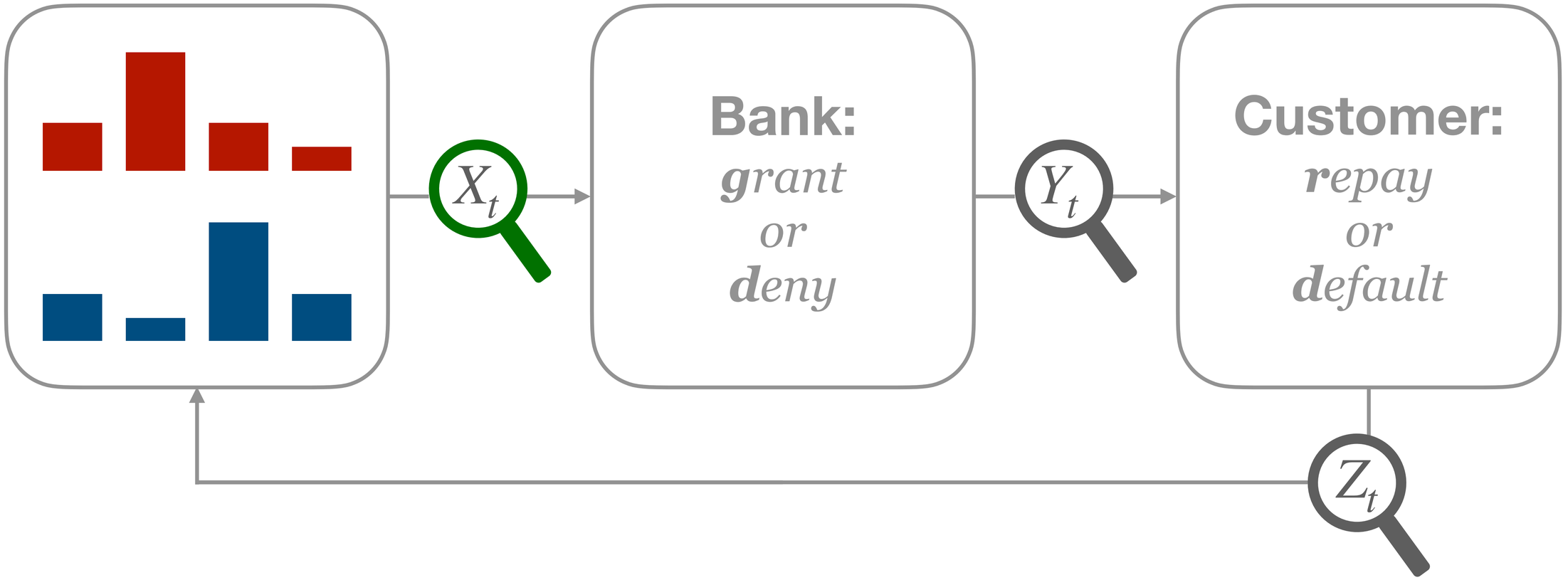
2.2 2.2 3.2

2.8 2.7 1.9



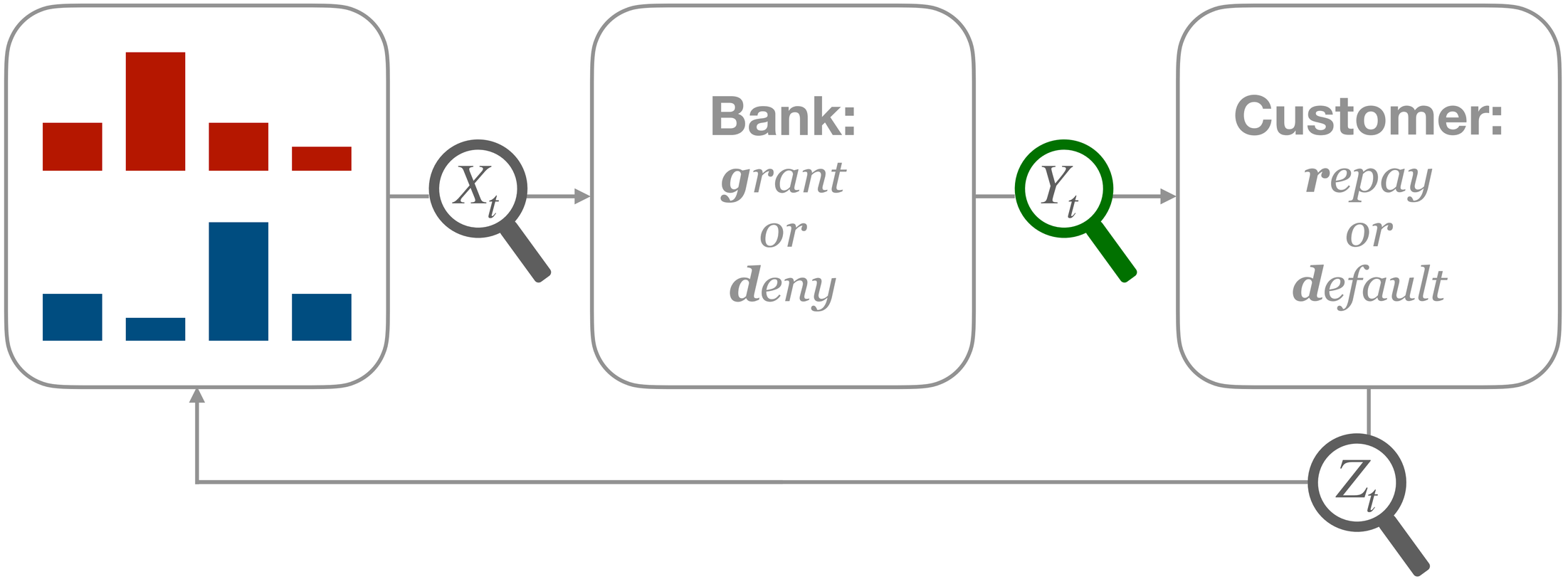
Time 1 *Time 2* *Time T*

Estimate the current disparity in average credit scores between Group Red and Group Blue



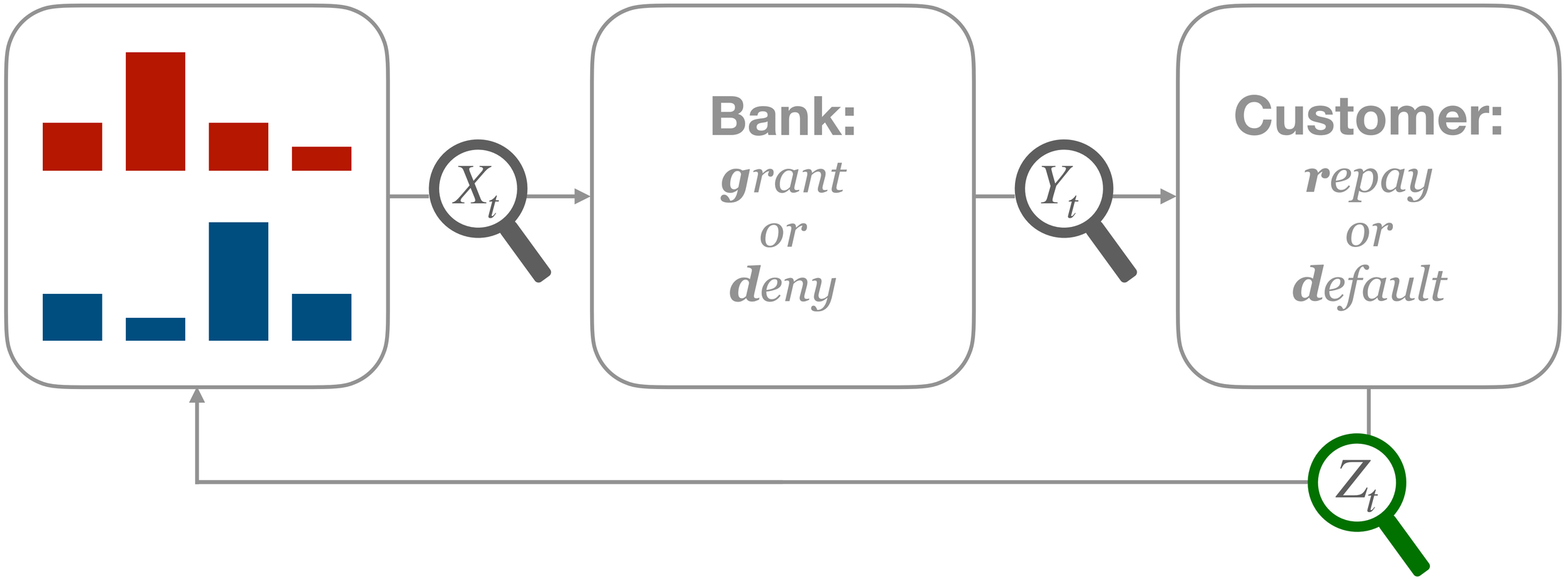
$$\vec{O}_t := O_1, \ldots, O_t = (X_1, Y_1, Z_1), \ldots, (X_t, Y_t, Z_t)$$

Sample



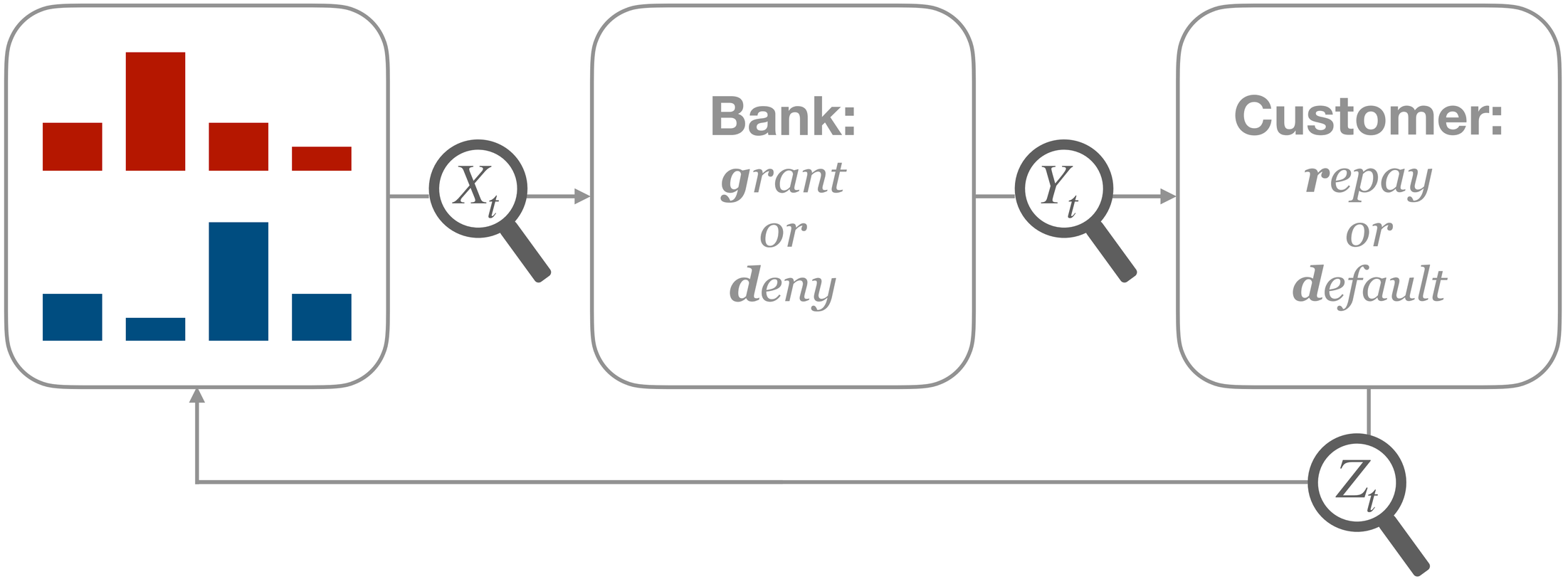
$$\vec{O}_t := O_1, \ldots, O_t = (X_1, Y_1, Z_1), \ldots, (X_t, Y_t, Z_t)$$

Decision



$$\vec{O}_t := O_1, \ldots, O_t = (X_1, Y_1, Z_1), \ldots, (X_t, Y_t, Z_t)$$

Z_1: *Outcome*



$$\varphi(\vec{o}_t) = \mathbb{E}_R(X_t \mid \vec{o}_{t-1}) - \mathbb{E}_B(X_t \mid \vec{o}_{t-1})$$

$\perp$

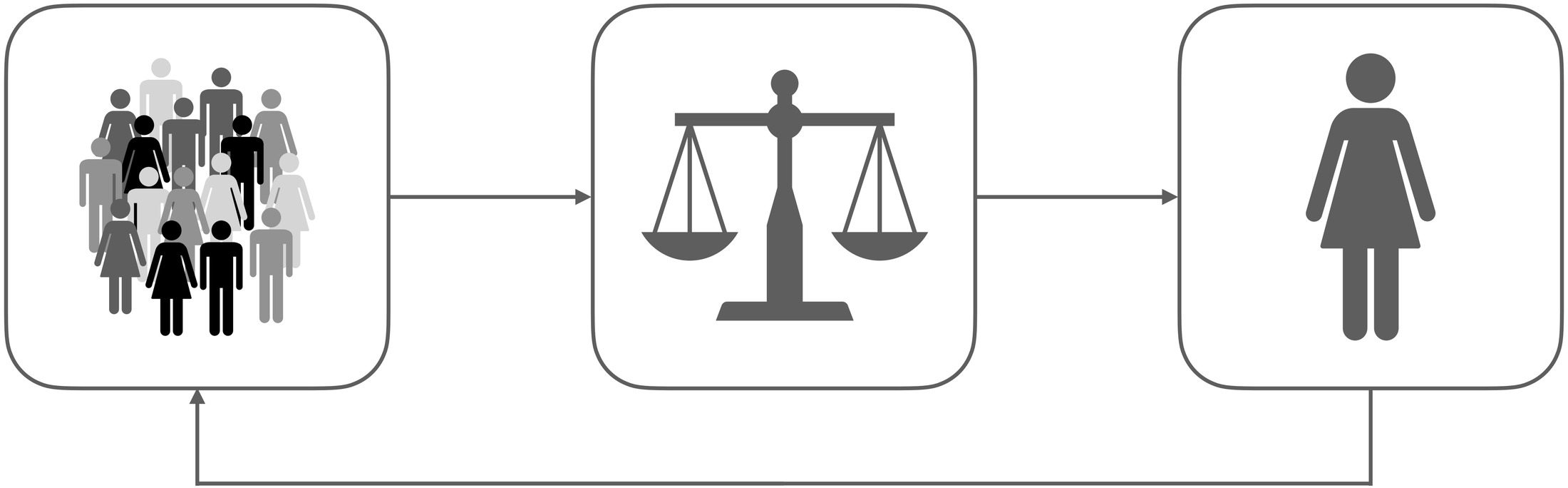
Past

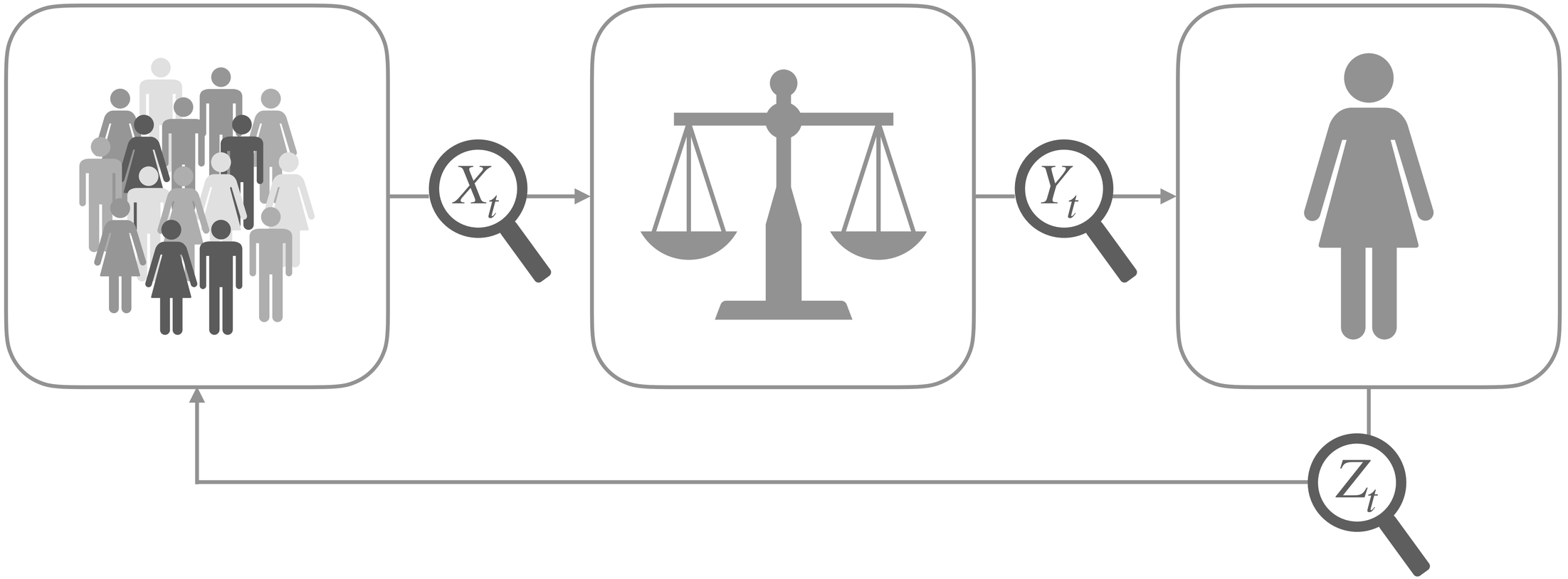
Goal.

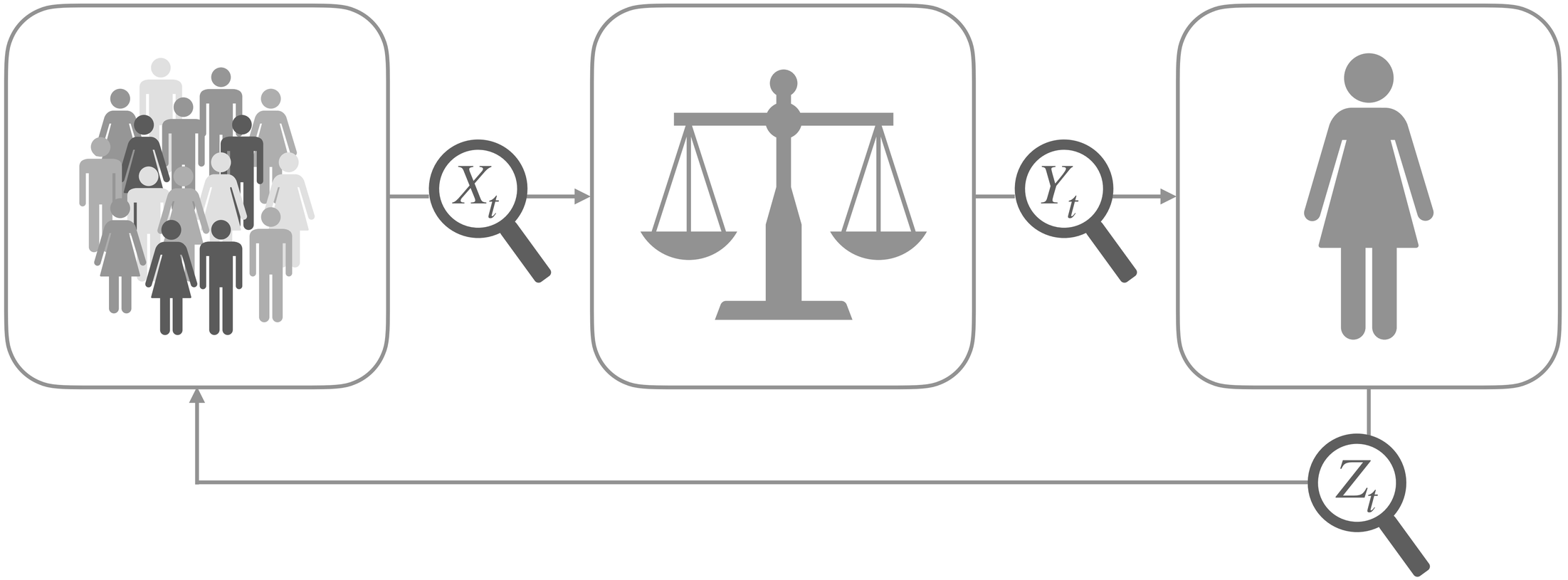
Design a program that watches the system and computes a confidence interval capturing the changing true value of the property φ with desired probability.

General Setting.

What assumptions are we making?

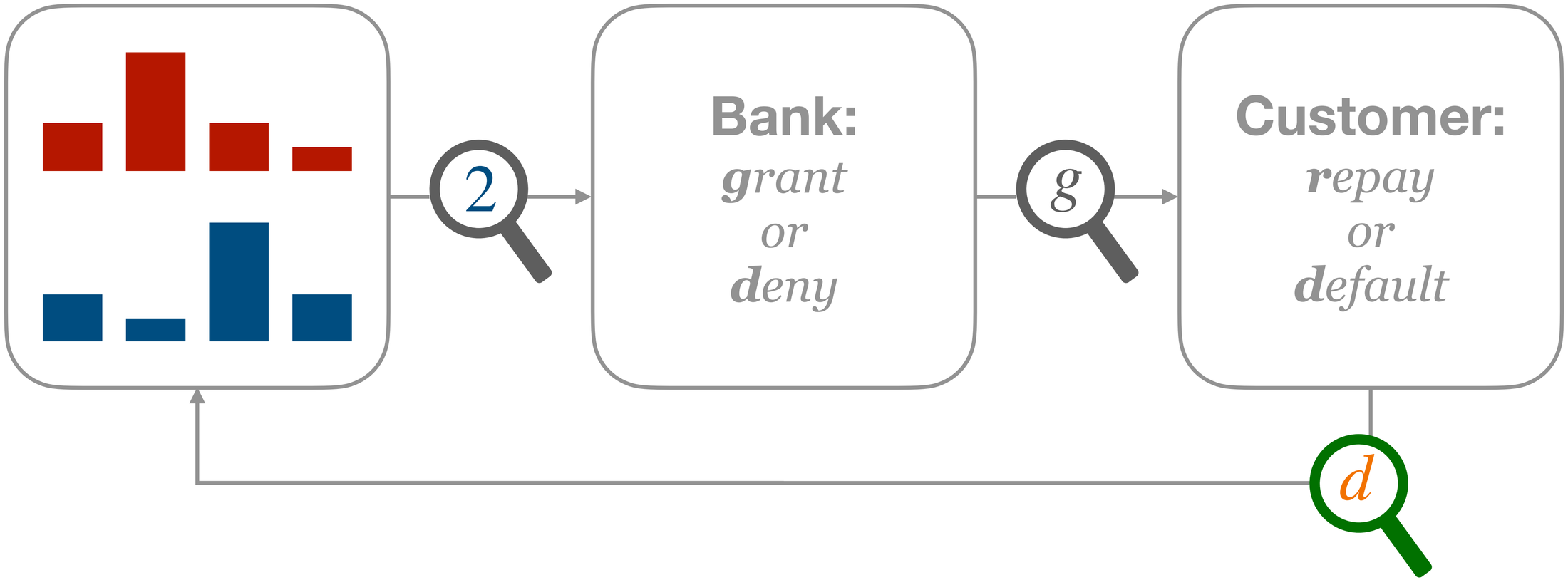



X_t
Y_t
Z_t



$$\mathbb{E}_G(X_{t+1} \mid \vec{o}_t) = \mathbb{E}_G(X_t \mid \vec{o}_{t-1}) + \Delta(o_t)$$

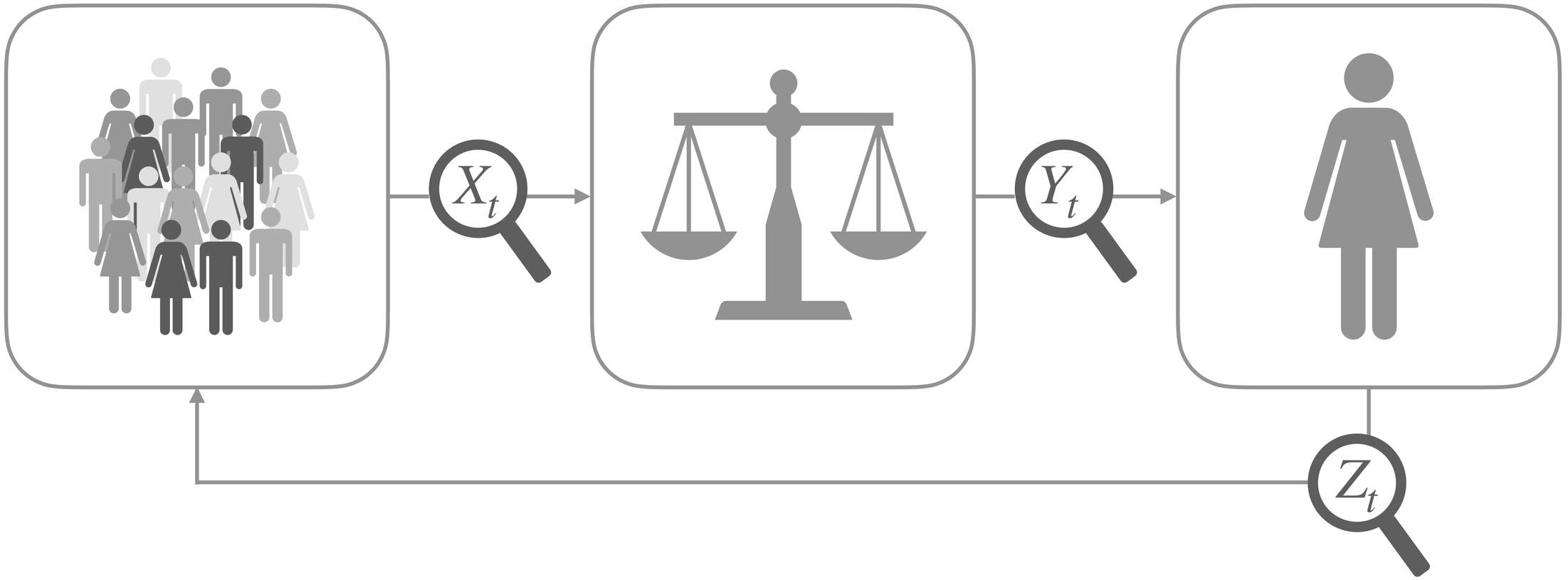
Change Function



$$\mathbb{E}_B(X_{t+1} \mid \vec{o}_t) = \mathbb{E}_B(X_t \mid \vec{o}_{t-1}) - \frac{1}{n_B}$$

Monitoring Problem.

What formal problem are we solving?



$$\varphi(o_t) = \mathbb{E}_{G_1}(\gamma(X_t) \mid y_t, z_t, \vec{o}_{t-1}) - \mathbb{E}_{G_2}(\gamma(X_t) \mid y_t, z_t, \vec{o}_{t-1})$$

Convex function

$$\mathbb{P}\left(\varphi(\vec{O}_t) \in \mathscr{M}(\vec{O}_t)\right) \geq 1 - \delta$$

Design a monitor $\mathscr{M}$ that given the observations $\vec{O}_t$ bounds the changing true value of the property φ with desired probability.

Algorithm.

Simple monitor with non-trivial soundness proof .

Estimate $\mathbb{E}_G(X_t \mid y_t, z_t, \vec{o}_{t-1})$ *for each group* G.

Estimate $\mathbb{E}_G(X_t \mid y_t, z_t, \vec{o}_{t-1})$ *for each group G.*

Compute confidence interval of estimates.

Estimate $\mathbb{E}_G(X_t \mid y_t, z_t, \vec{o}_{t-1})$ *for each group* G.

Compute confidence interval of estimates.

Push confidence intervals through $\gamma(\cdot)$.

Estimate $\mathbb{E}_G(X_t \mid y_t, z_t, \vec{o}_{t-1})$ *for each group* G.

Compute confidence interval of estimates.

Push confidence intervals through $\gamma(\cdot)$.

Apply union bound (and interval arithmetic) to compute confidence interval of the property.

Confidence Interval.

Doob-Martingales and Azuma's Inequality

Estimator of $\mathbb{E}(X_1)$

$$\mathbb{P}\left(\left|\mathbb{E}(X_1) - \hat{E}_1(\overrightarrow{O}_t)\right| \geq \varepsilon\right) \leq \delta$$

Bound estimator of $\mathbb{E}(X_1)$

$$\mathbb{E}\left(\hat{E}_1(\vec{O}_t)\right), \mathbb{E}\left(\hat{E}_1(\vec{O}_t)\middle|\vec{O}_1\right), \ldots, \mathbb{E}\left(\hat{E}_1(\vec{O}_t)\middle|\vec{O}_t\right)$$

Doob-Martingale

$$\mathbb{E}\left(\hat{E}_1(\overrightarrow{O}_t)\middle|\overrightarrow{O}_{k+1}\right) - \mathbb{E}\left(\hat{E}_1(\overrightarrow{O}_t)\middle|\overrightarrow{O}_k\right)$$

Bound Difference

$$\mathbb{P}\left(\left|\mathbb{E}\left(\hat{E}_1(\vec{O}_t)\right) - \mathbb{E}\left(\hat{E}_1(\vec{O}_t)\middle|\vec{O}_t\right)\right| \geq \varepsilon\right) \leq \delta$$

Azuma's inequality

$$\mathbb{E}(X_1)$$

$$\mathbb{P}\left(\left|\mathbb{E}\left(\hat{E}_1(\vec{O}_t)\right) - \mathbb{E}\left(\hat{E}_1(\vec{O}_t)\middle|\vec{O}_t\right)\right| \geq \varepsilon\right) \leq \delta$$

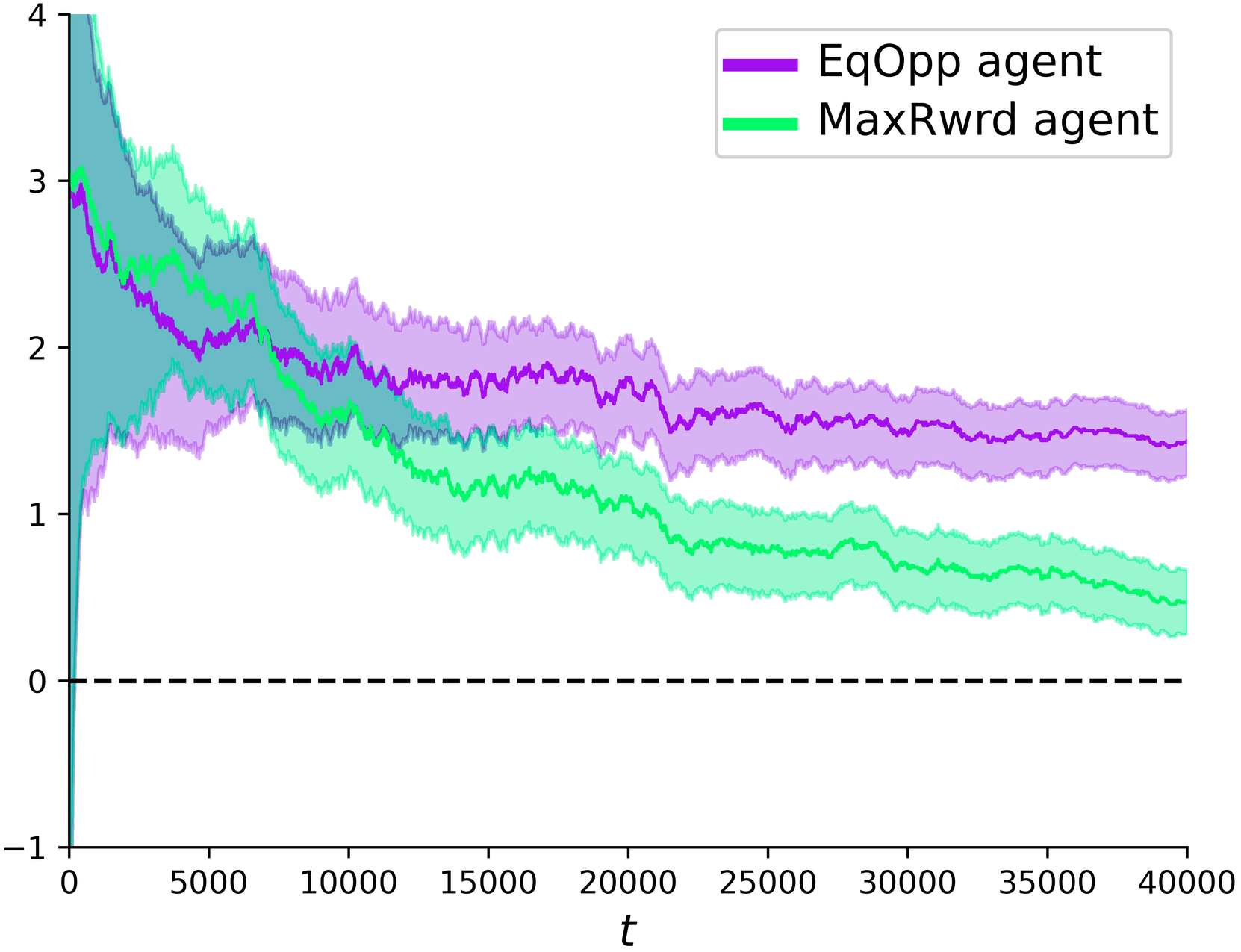
Azuma's inequality

$$\mathbb{P}\left(\left|\underbrace{\mathbb{E}\left(\hat{E}_1(\vec{O}_t)\right)}_{\mathbb{E}(X_1)} - \underbrace{\mathbb{E}\left(\hat{E}_1(\vec{O}_t)\middle|\vec{O}_t\right)}_{\hat{E}_1(\vec{O}_t)}\right| \geq \varepsilon\right) \leq \delta$$

Azuma's inequality

Experiments.

Lending and Attention (D'Amour 2020).


EqOpp agent
MaxRwrd agent
4
3
2
1
0
−1
φ_t
0
5000
10000
15000
20000
25000
30000
35000
40000
t

Related Work.

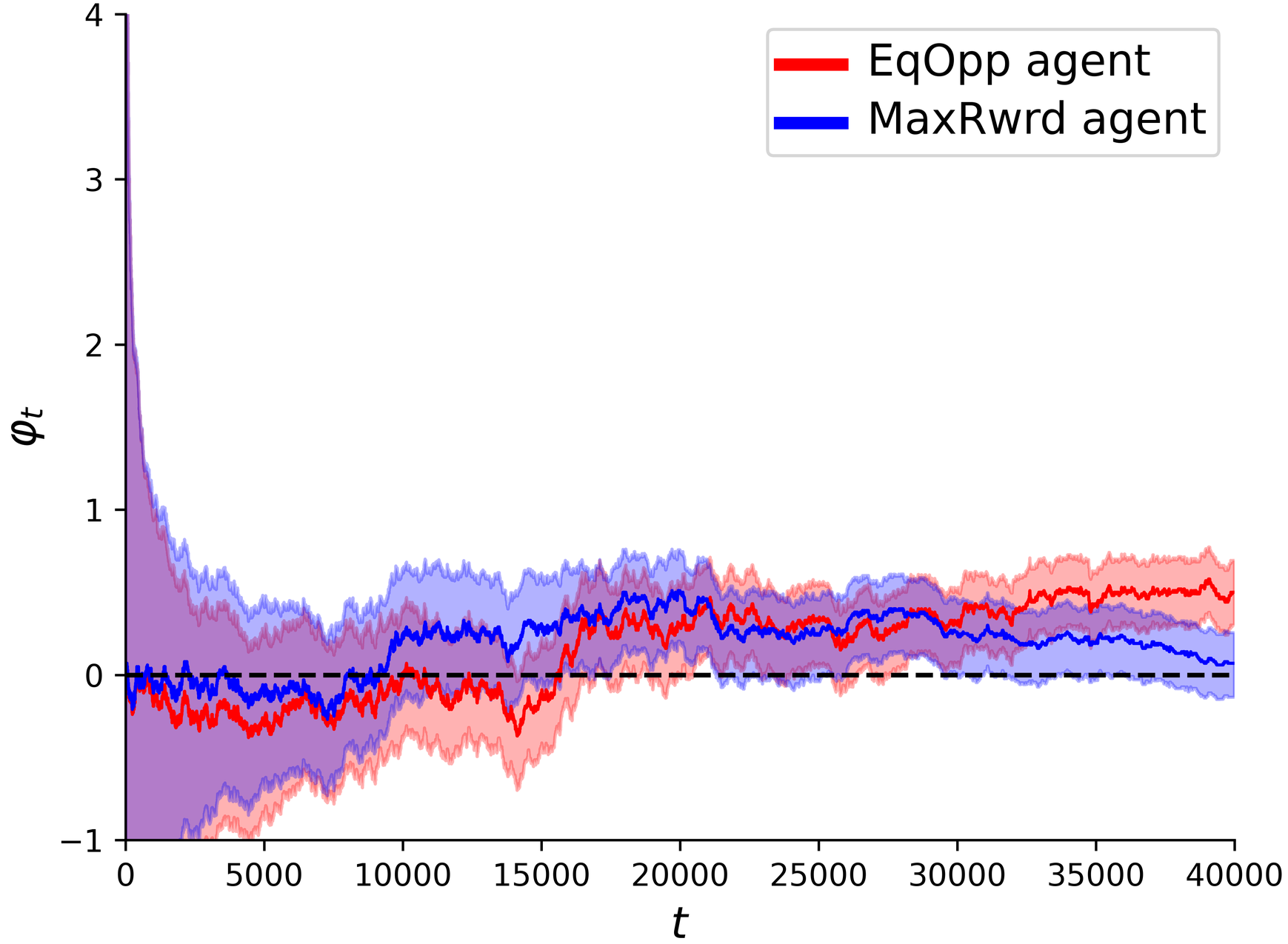
Pioneered by (Albarghouthi 2019).
Extended to Markov chains (Henzinger 2023).
Problems in dynamic fairness (D'Amour 2020).

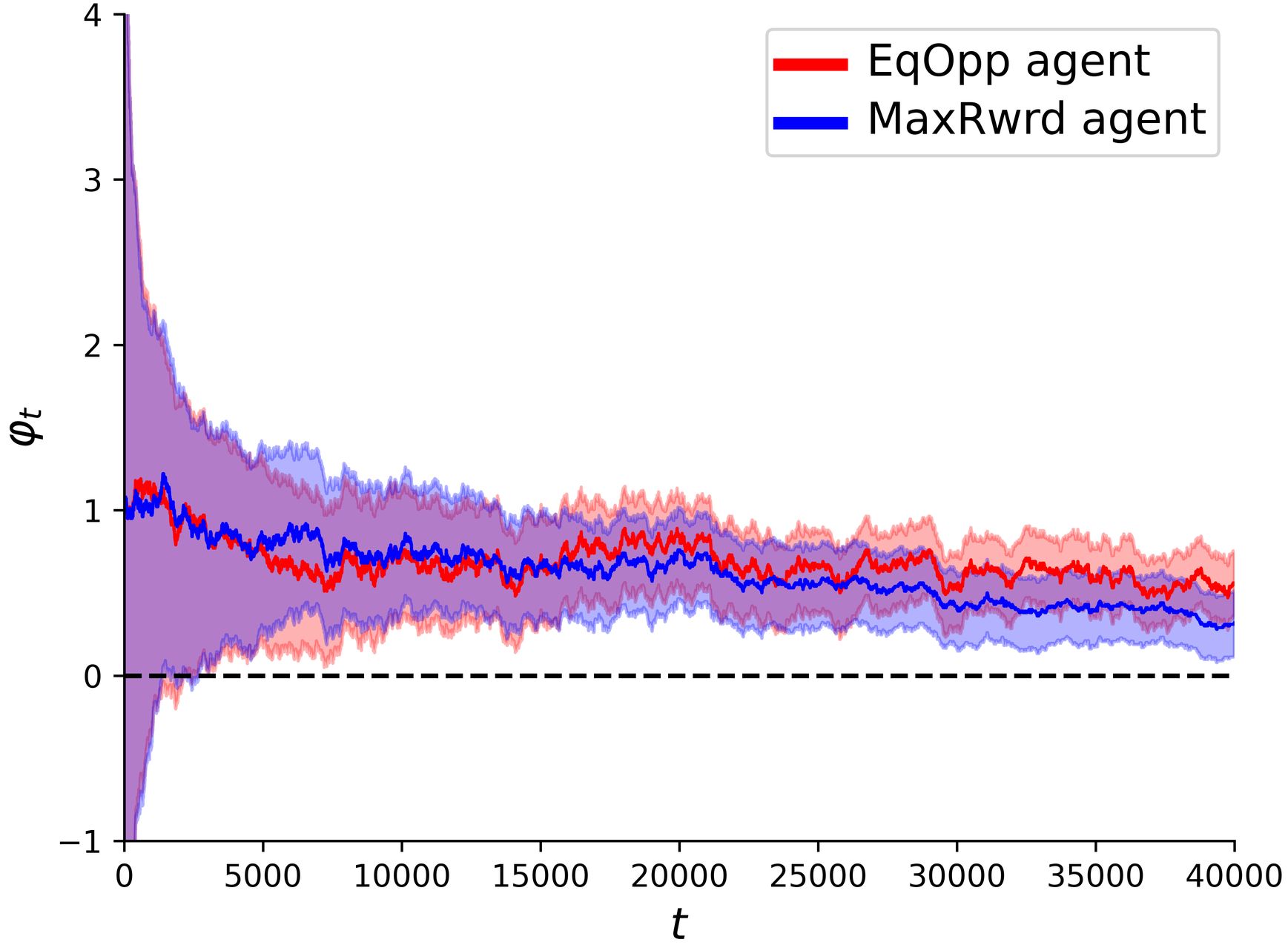
Aws et al. 2019. Fairness-aware programming
D'Amour et al. 2020. Fairness is not static: deeper understanding of long term fairness via simulation studies
Henzinger, et al. Monitoring algorithmic fairness.

Conclusion.

Detecting unfair behaviour in deployed systems using simple light-weight monitors.
How can we correct for unfair behaviour?

Institute of Science and Technology Austria


EqOpp agent
MaxRwrd agent
4
3
2
1
0
−1
φ_t
0
5000
10000
15000
20000
25000
30000
35000
40000
t


EqOpp agent
MaxRwrd agent
4
3
2
1
0
−1
φ_t
0
5000
10000
15000
20000
25000
30000
35000
40000
t