運用の効率化に向けた通知の課題とZabbix での対処法

Zabbix Conference Japan 2023

2023/11/16

SRA OSS 合同会社

ZABBIX PREMIUM PARTNER

Who am I

- 赤松 俊弘（Toshihiro Akamatsu）
  - SRA OSS合同会社
  - Zabbix 認定プロフェッショナル
  - Zabbix 歴 7 年

- 業務
  - OSS 全般の技術サポート、構築、コンサルティング
  - 主に Zabbix を担当

監視における通知と運用

- 監視運用者にとって、毎日は通知との闘い
  - 日々大量の通知が届く
  - 大量の通知から優先度・重要度を判断して対応する必要がある

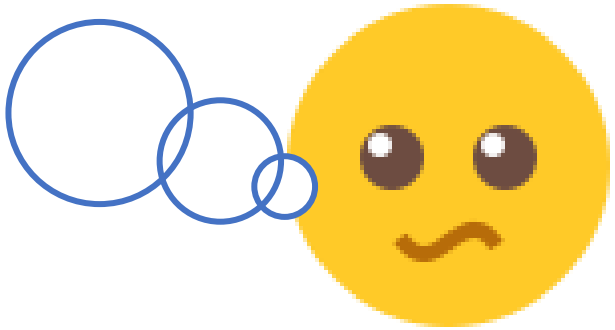
- あまりにも通知が多いと…
  - 緊急・重要な通知を見逃す
  - そもそも通知への対応がおざなりになる

監視運用者の願い

- ノイズとなる通知はできるだけ減らしたい
- 通知対応を効率的にしてコストを減らしたい

でも

- Zabbix でどう設定したらよいか分からない



本講演の目的

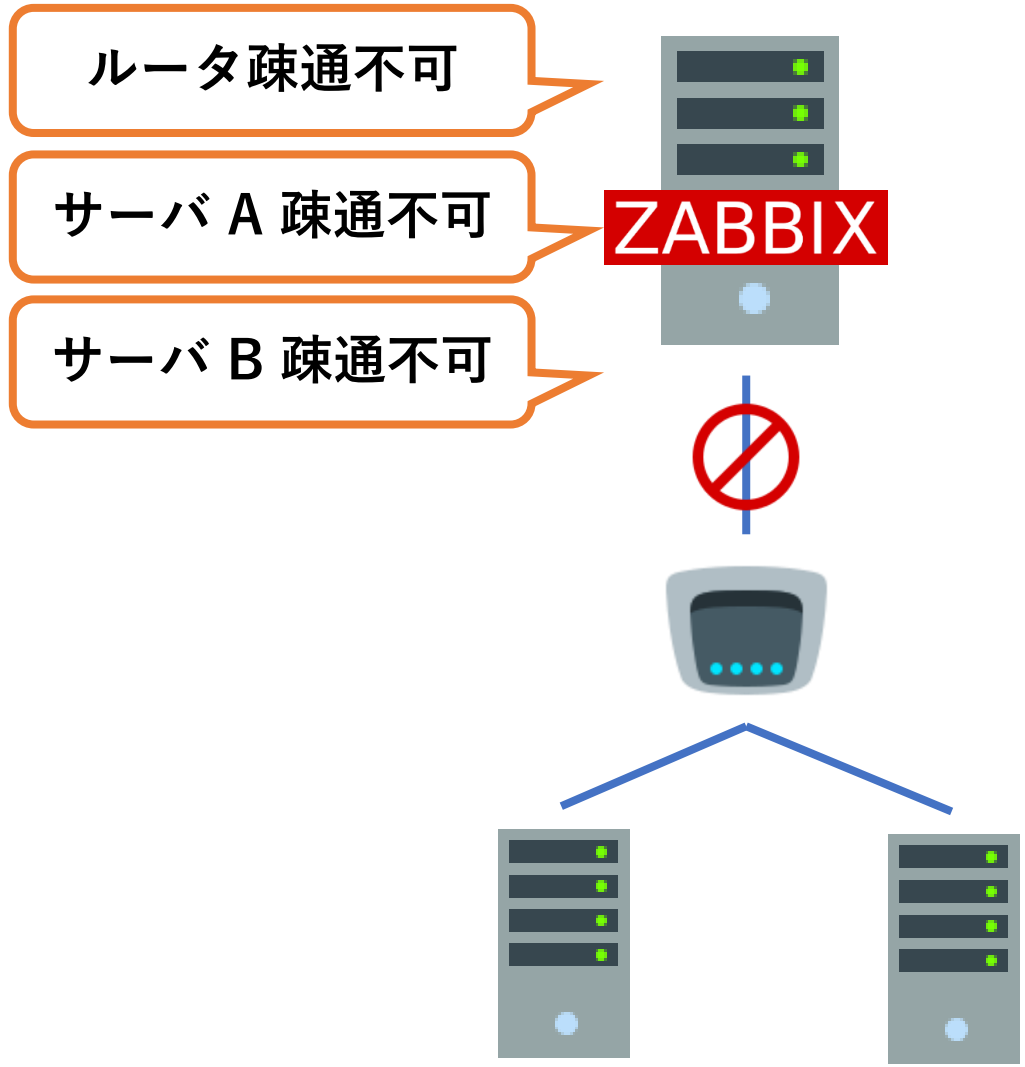
Zabbix でできる
効率的な障害検知・通知の
やり方Tips を紹介

Case①

障害の波及による障害多発

障害が波及するケース

上位の機器の障害で
下位の機器まで障害判定

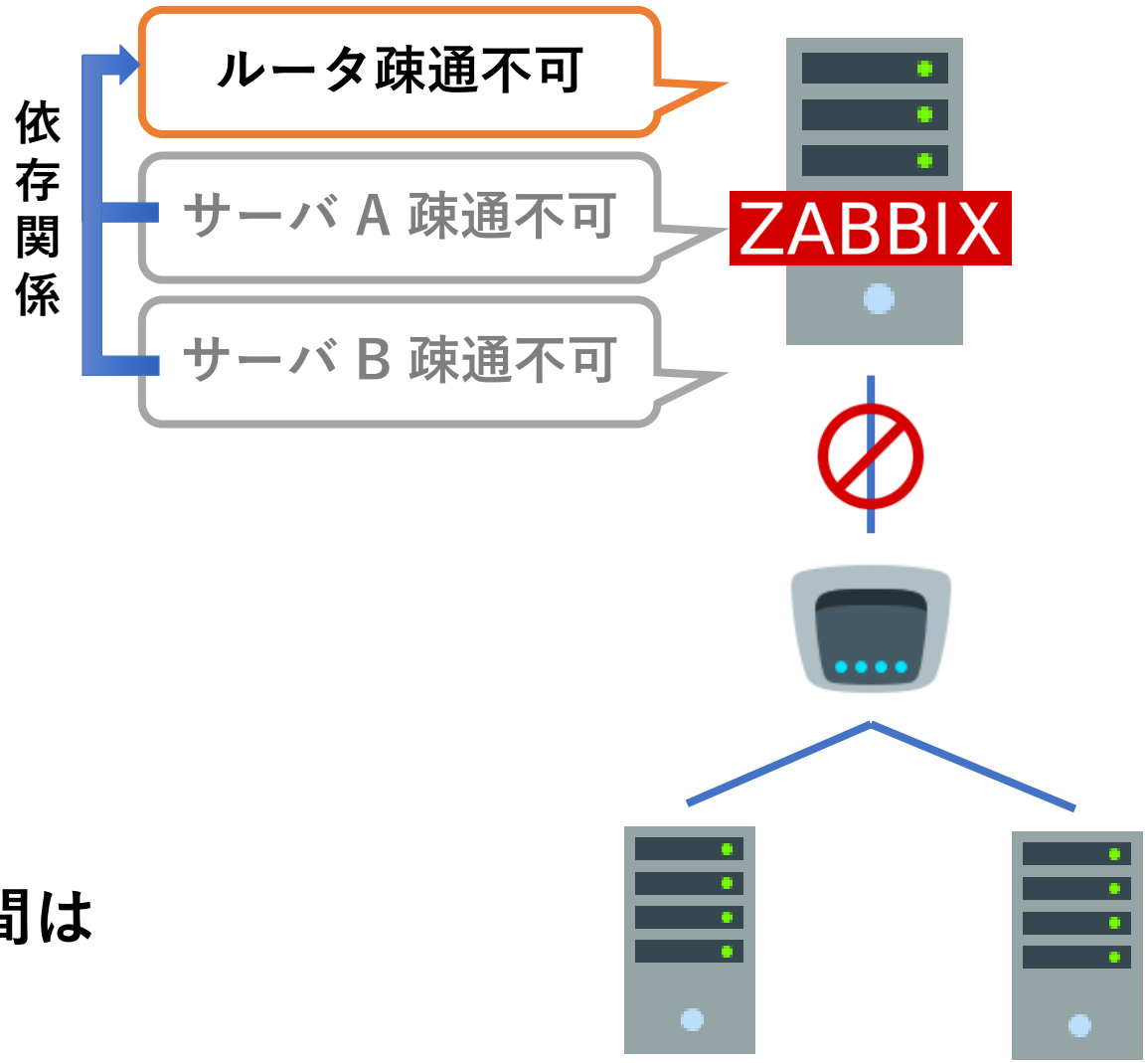


障害が波及するケースの改善策

上位の機器の障害で
下位の機器まで障害判定

トリガーの依存関係で
下位の障害を抑制

上位（依存先）のトリガーが障害の間は
下位のトリガーを障害としない



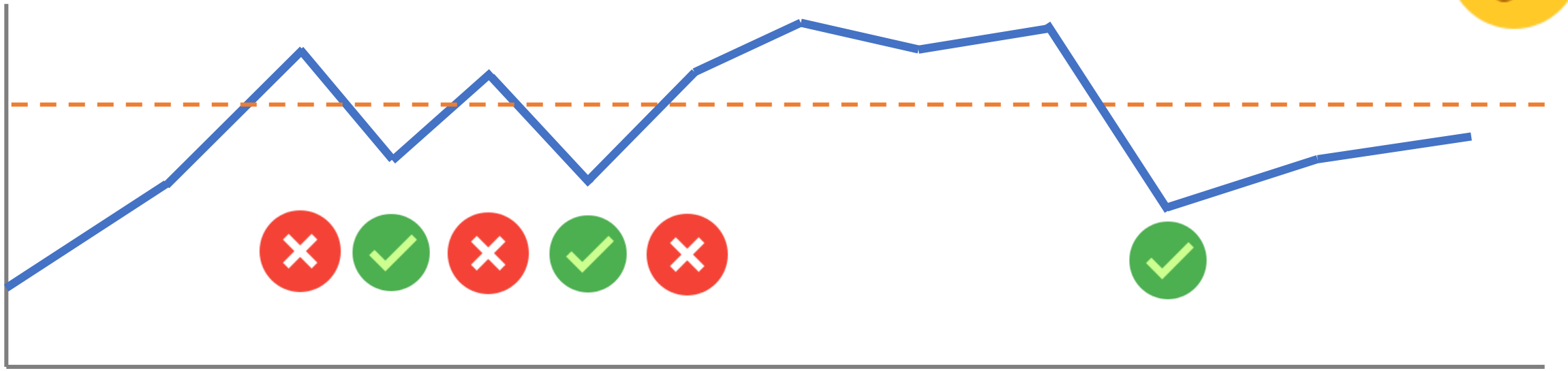
Case②

数値監視での障害の多発

数値監視でのケース

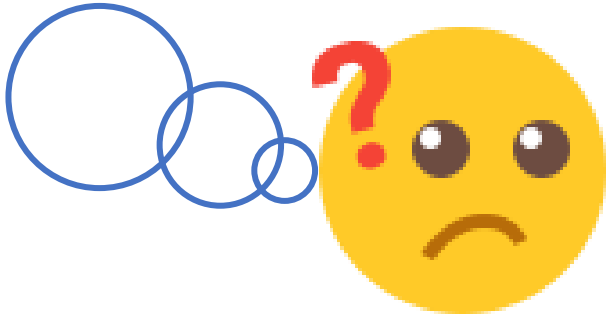
- last(/host/key) > X
- 最新値が閾値 X を超えているかどうかのみで判定

短時間で閾値を連続でまたぐ場合、障害・復旧通知がその都度届く
突発的な負荷上昇（スパイク）でも検知して通知してしまう



数値監視での改善策

- last 関数が問題？
- 他の関数やトリガー設定次第でどうにかなる？



トリガーによる改善策：max関数

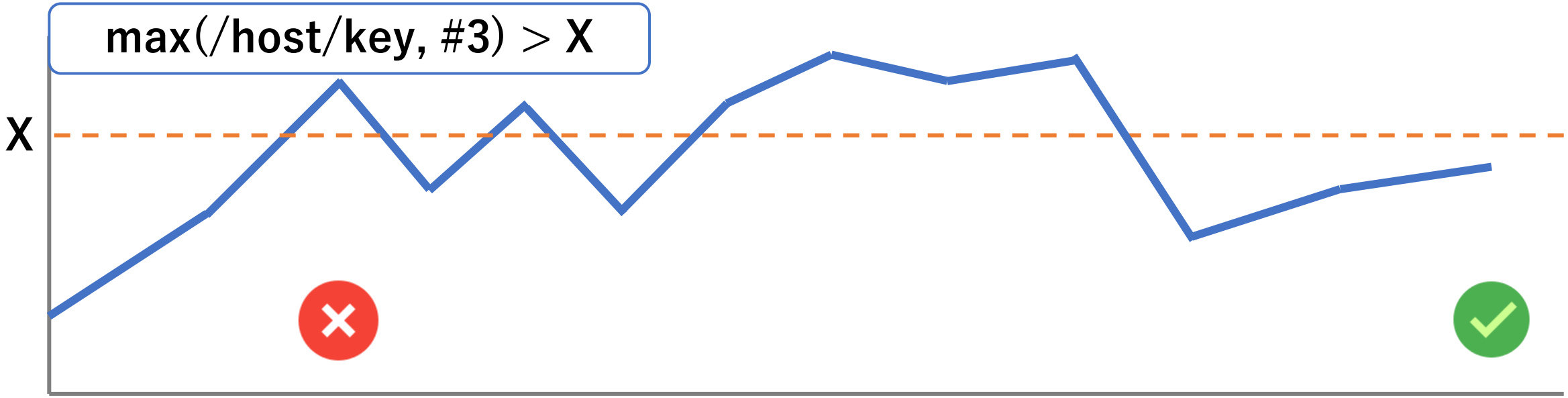
設定

max(/host/key, #N) > X

過去 N 回の最大値が閾値 X を
超えると障害

特徴

- 閾値超えで即障害
- 一定期間閾値を下回れば復旧
- 閾値前後の動き：抑制
- スパイク：検知



トリガーによる改善策：min関数

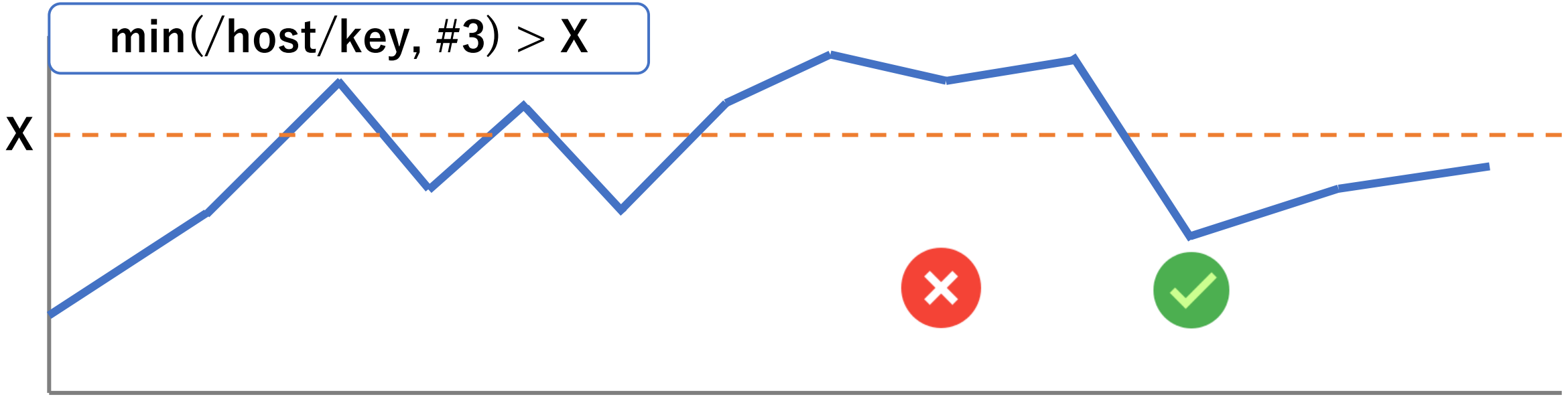
設定

min(/host/key, #N) > X

過去 N 回の最小値が閾値 X を
超えると障害

特徴

- 一定期間閾値超えで障害
- 閾値を下回れば即復旧
- 閾値前後の動き：抑制
- スパイク：抑制



トリガーによる改善策：count関数

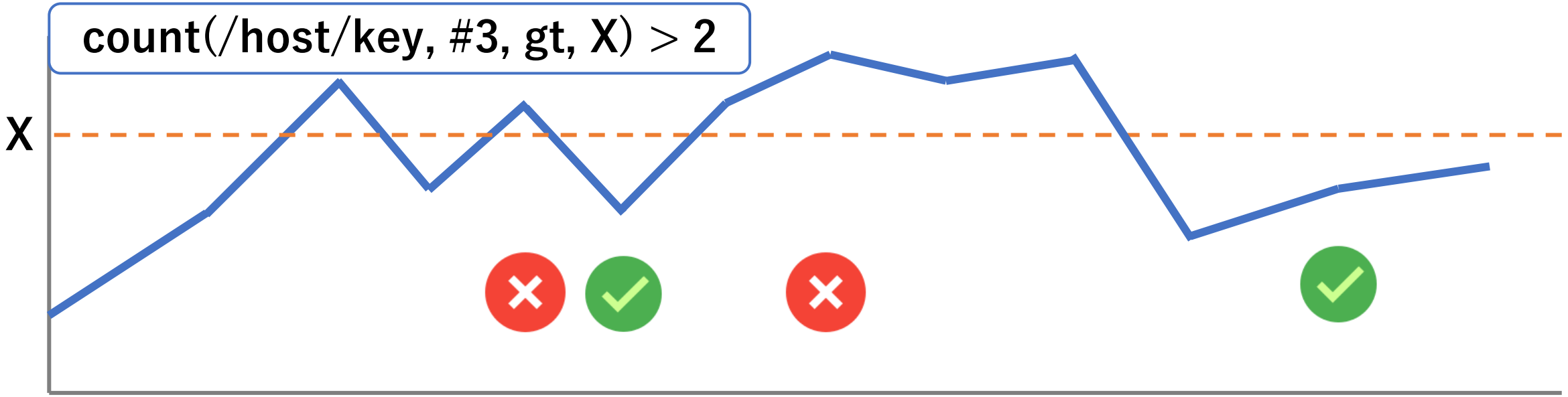
設定

count(/host/key, #N, gt, X) > C

過去 N 回で閾値 X を超えた回数が C を超えると障害

特徴

- 一定回数閾値超で障害
- 一定回数閾値を下回ると復旧
- 閾値前後の動き：ある程度抑制
- スパイク：抑制



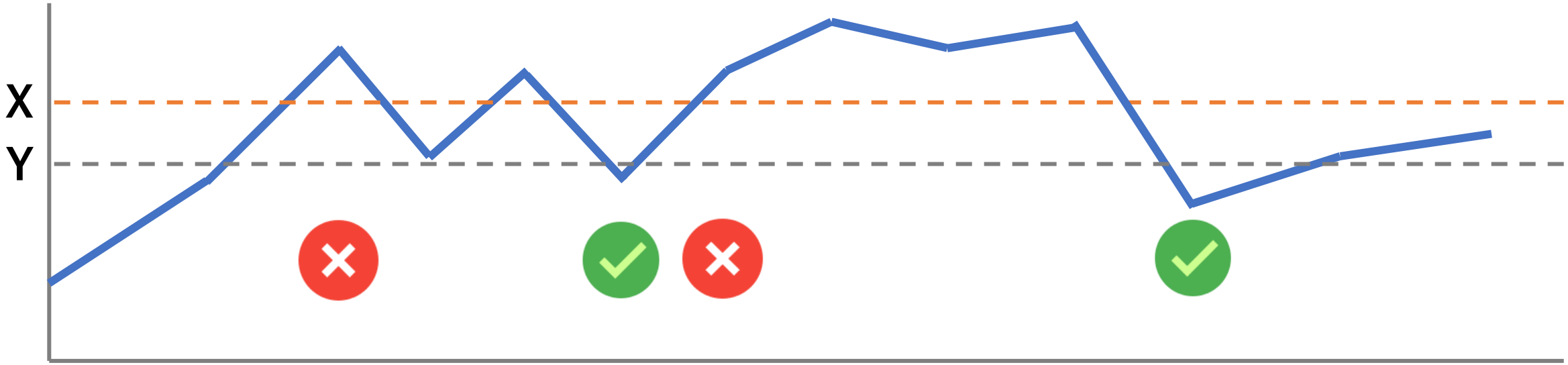
トリガーによる改善策：復旧条件式

設定

障害：last(/host/key) ＞ X
復旧：last(/host/key) ＜ Y
最新値が閾値 X を超えると障害
閾値 Y を下回ると復旧

特徴

- 閾値超で即障害
- 閾値を下回れば即復旧
- 閾値前後の動き：ある程度抑制
- スパイク：検知

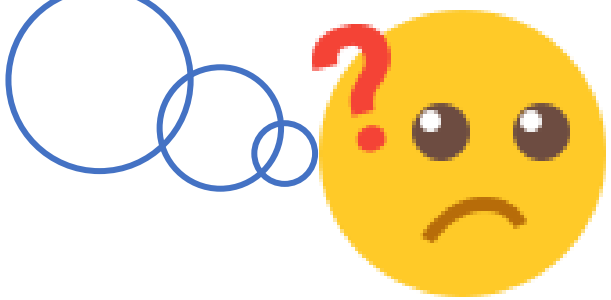


トリガーによる改善策まとめ

改善策	障害感度	復旧感度	短期間の閾値またぎ	スパイク	備考
max関数	○	△	○	×	感度は期間設定による
min関数	△	○	○	○	感度は期間設定による
count関数	△	△	△	○	感度と閾値またぎは期間と回数による
復旧条件式	○	○	△	×	閾値またぎは閾値の設定による

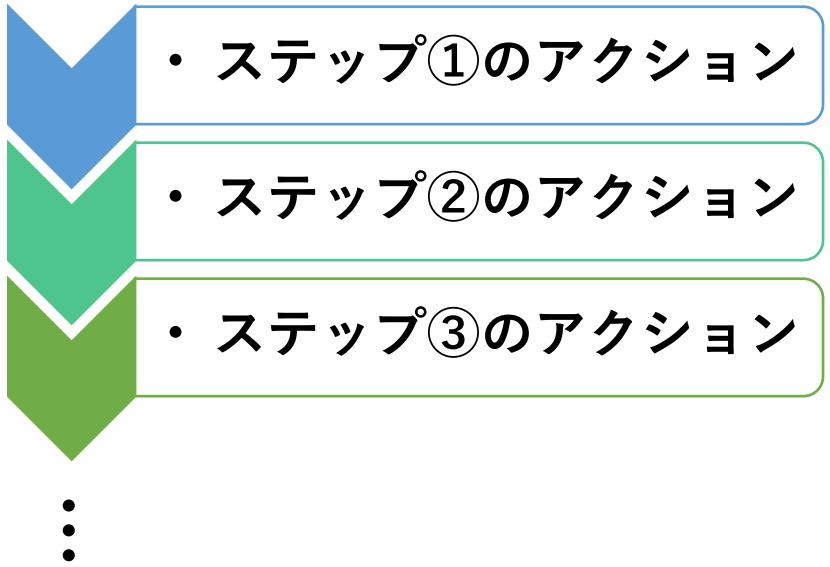
数値監視でのトリガー以外の改善策

- トリガー以外で対策は？
- 通知の問題だから
  アクション設定だとどう？



アクションによる改善策

- Zabbix のアクションの
  エスカレーション機能を利用
- 段階（ステップ）を踏んで
  異なるアクションを実行



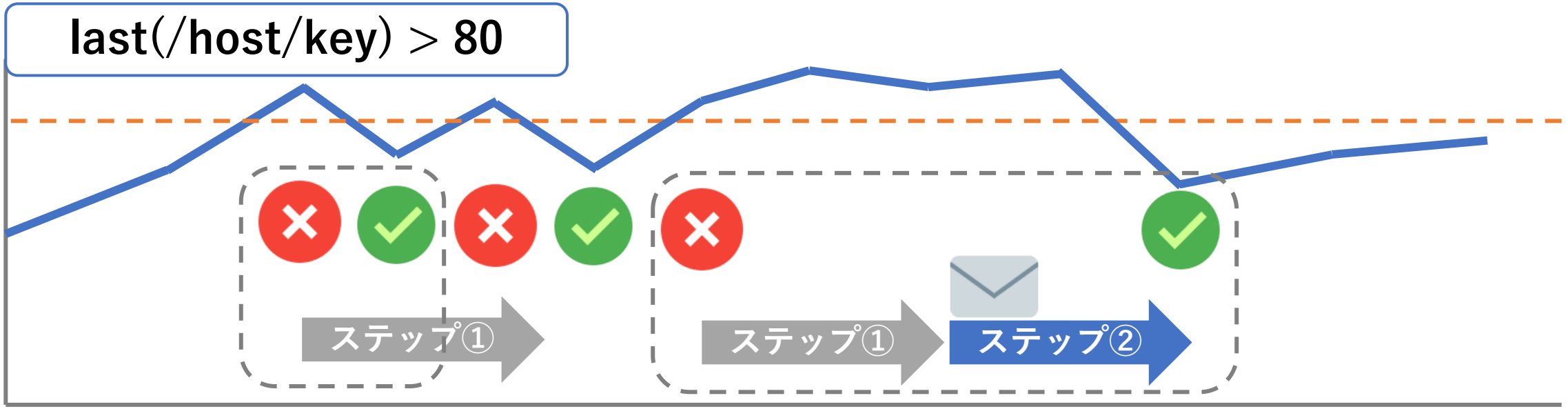
アクションによる改善策

設定

- ステップ間隔：N（監視間隔以上）
- ステップ①は通知なし
- ステップ②で通知

特徴

- 一定期間障害で通知
- 障害通知なしなら復旧通知なし
- 閾値前後の動き通知：抑制
- スパイク通知：抑制



数値監視の改善策まとめ

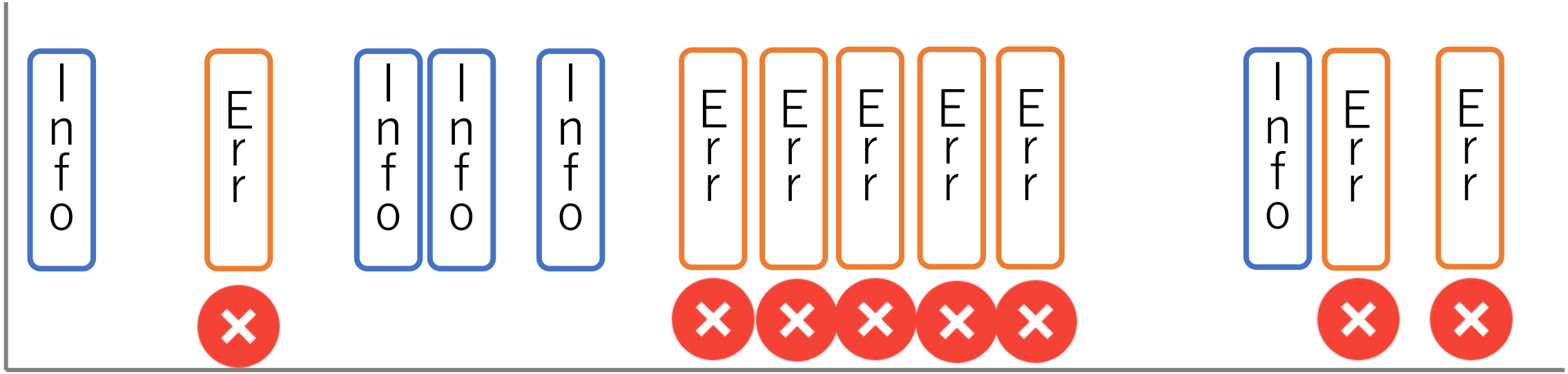
障害通知の多少の遅延を許容できるなら、
min 関数もしくはアクションによる改善策

Case③

ログ・SNMP トラップでの障害の多発

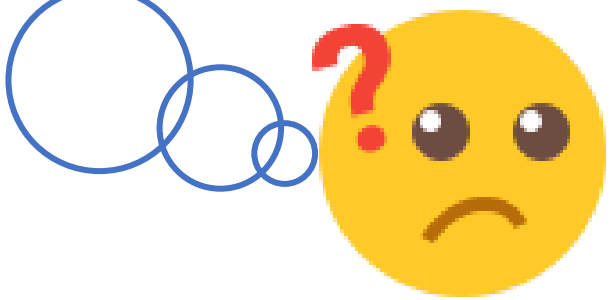
ログ・SNMP トラップでのケース

- find(/host/log[/var/log/messages, error],, like, error) = 1
- 検知文字列が出力されると障害
- 短時間に大量のログ・トラップが出力されると
  そのたびに通知が届く



ログ・SNMP トラップ監視での改善策

- 短期間の件数が多ければ
  障害にしなければ
  いいのでは？



トリガーによる改善策：count関数

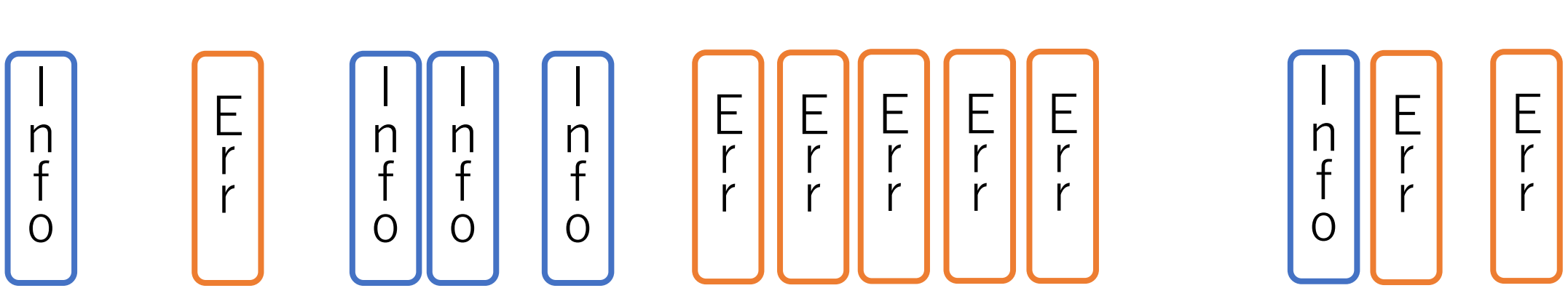
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

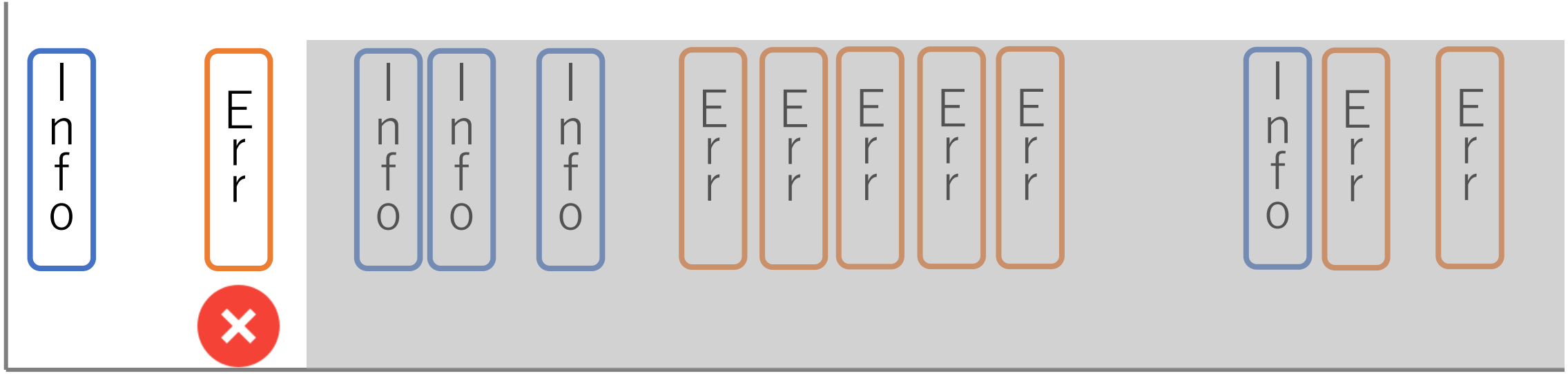
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

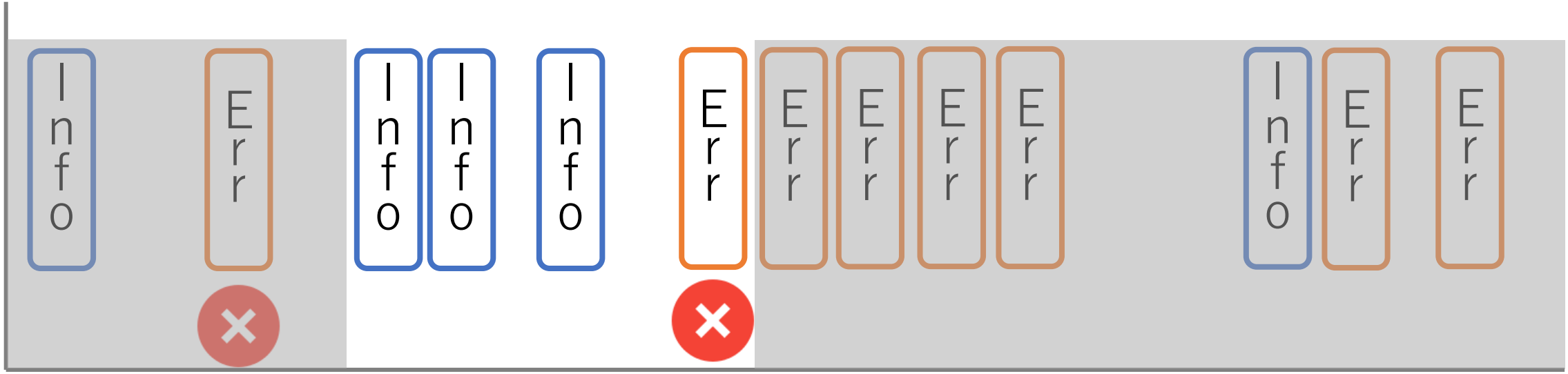
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

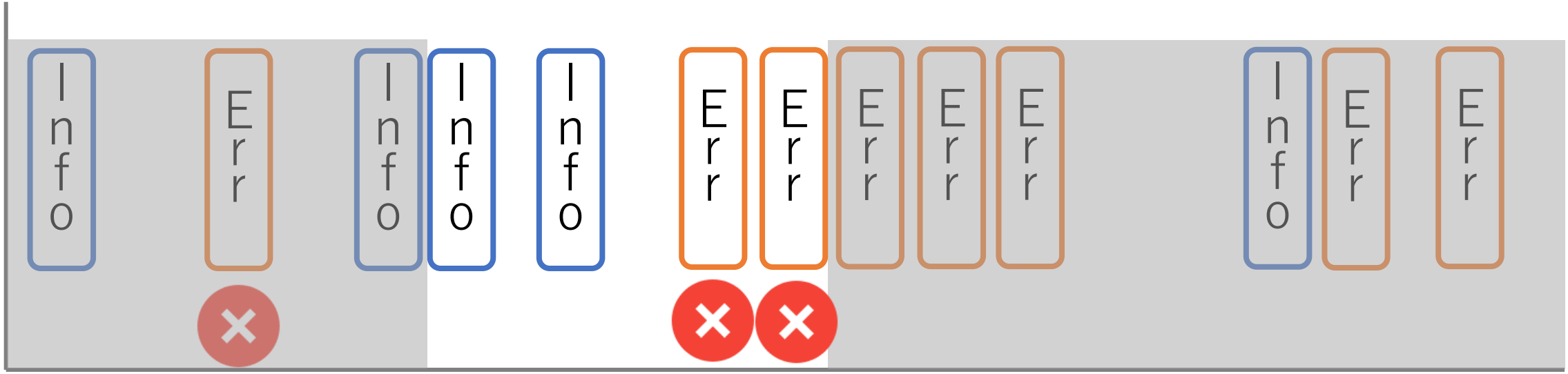
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

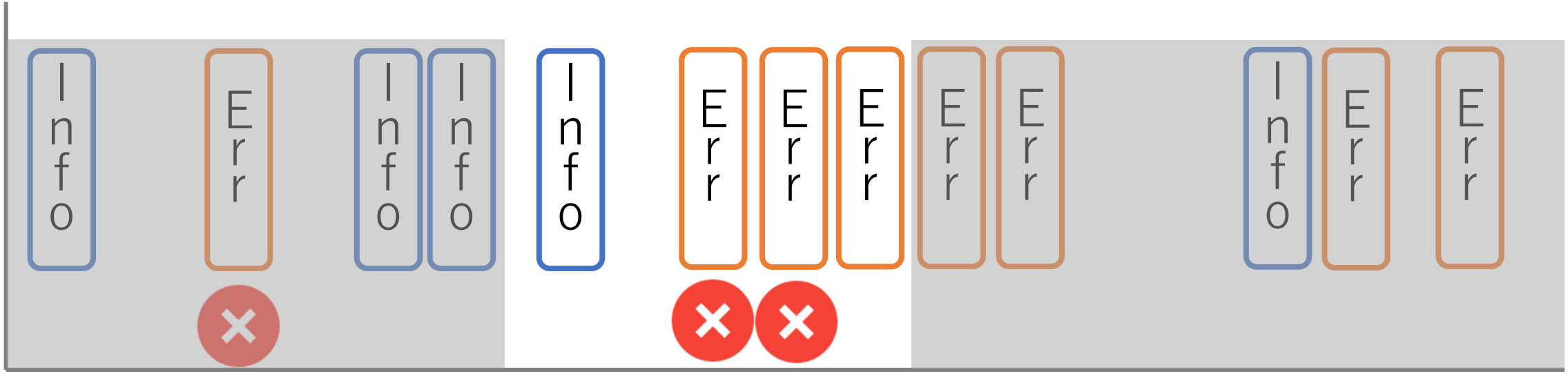
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

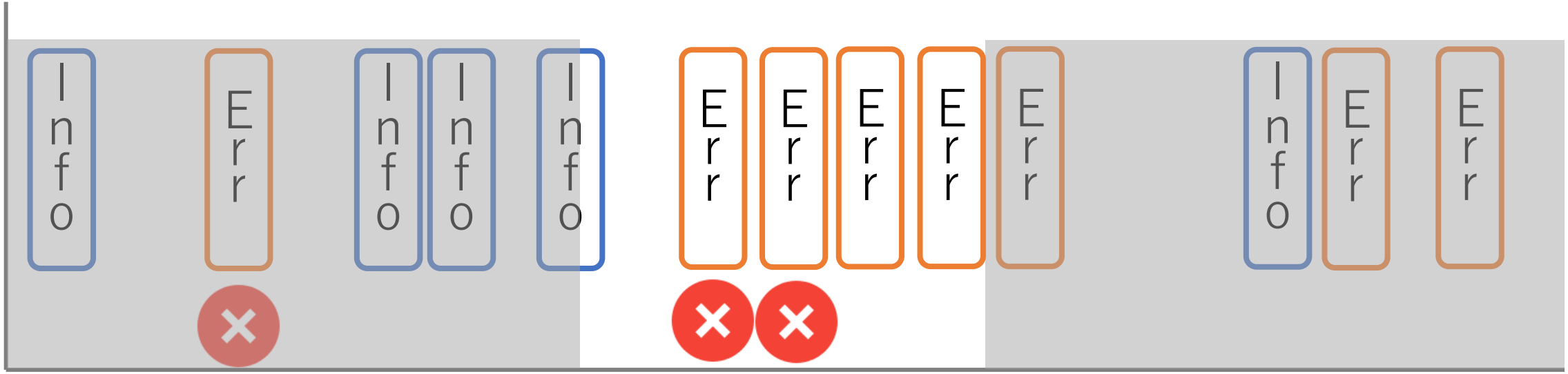
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

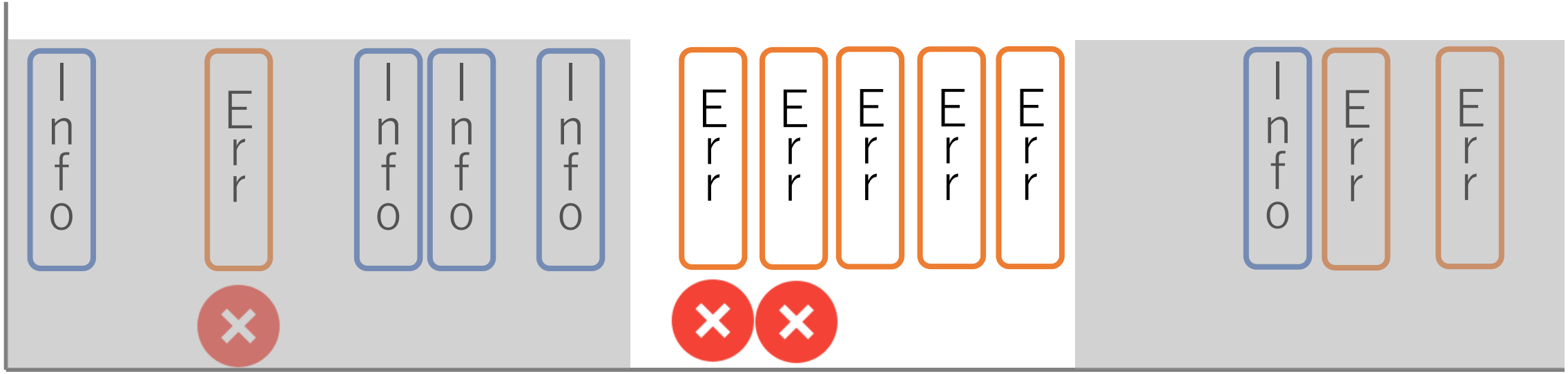
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：count関数

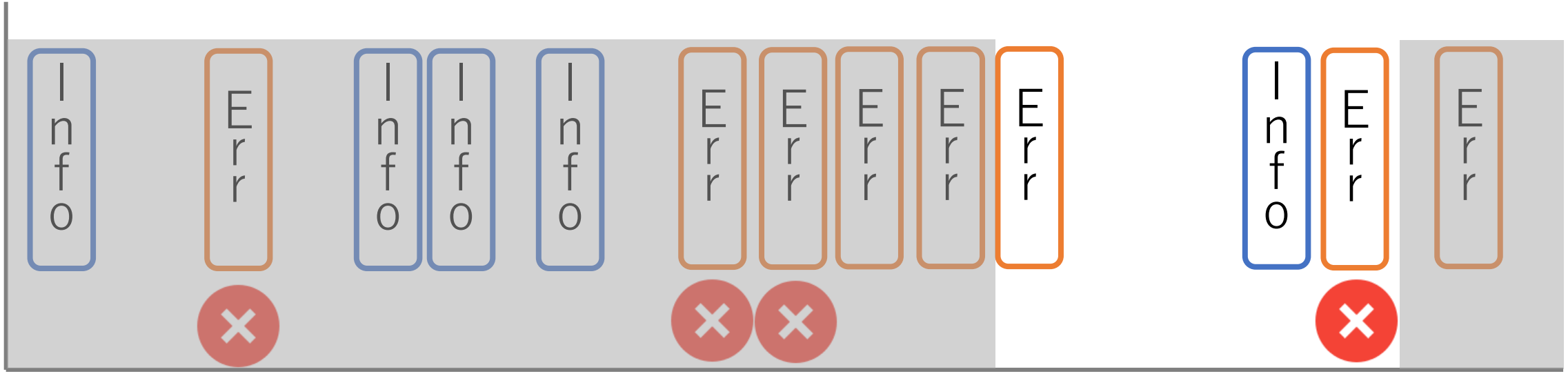
設定

find(/host/log[file, str],, like, str) = 1 and
count(/host/log[file, str], T, like, str) < N

最新値に str を含むかつ
過去 T 時間に str を含む件数が N 件未満で障害

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると無視
- いつ大量出力があったかは分からない



トリガーによる改善策：依存関係

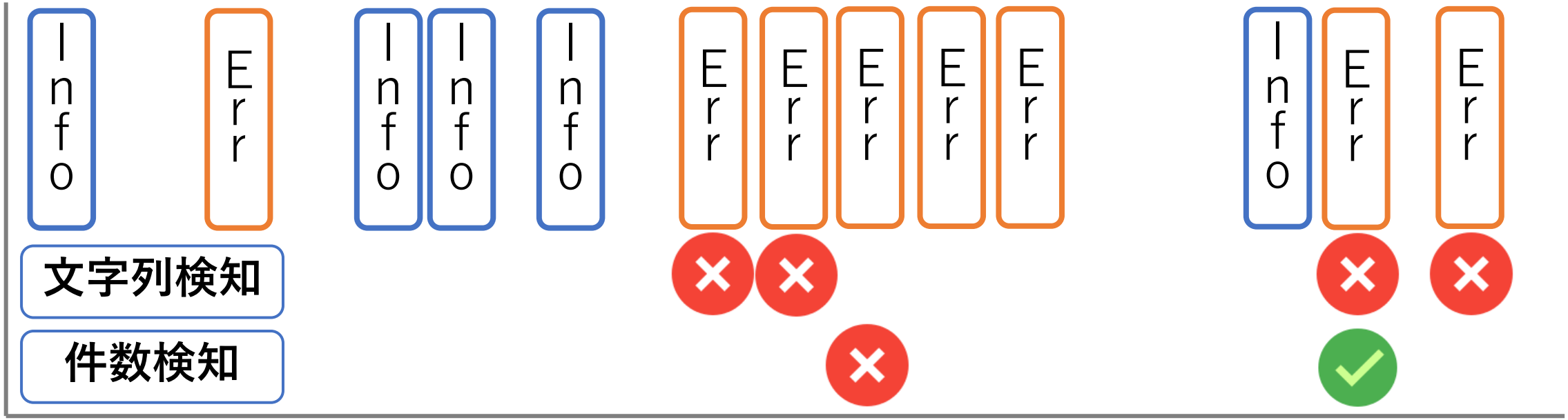
設定

find(/host/log[file, str],, like, str) = 1
依存先:sum(/host/log.count[file, str], T) > N

最新値に str を含むと障害とするが、
過去 T 時間に str を含む件数が N 件超で抑制

特徴

- 通常は検知文字列で障害
- 短期間に大量に出ると抑制
- 大量出力があったことは log.count で検知



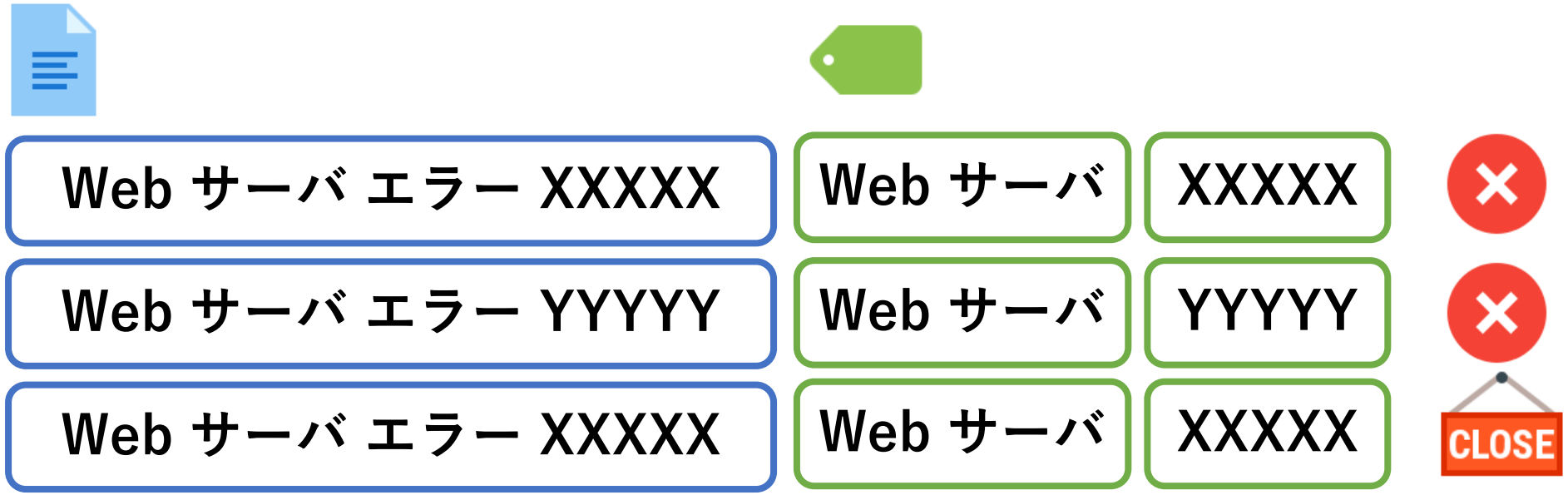
イベント相関関係による改善策

設定

イベント相関関係：
古いイベントタグと新しいイベントタグが
等しい場合、新しいイベントをクローズ

特徴

- 障害中のイベントと
同じタグのイベントは
自動クローズ
- 最初の障害が復旧しないと
新しい障害は検知されない



nodata 関数を使った復旧

- 一定時間検知文字列が出力されなければ復旧させたい

設定
障害：find(/host/log[file, str], T, like, str) = 1 復旧：nodata(/host/log[file, str], T) =1

- 復旧させたい時間幅 T を障害条件式にもいれておく
- 障害条件式が False にならないと復旧条件式が評価されない

ログ・SNMP トラップ監視の改善策まとめ

大量出力中であることを検知したいなら
log.count との複合技、
内容によって細かく制御したいなら
イベント相関関係

他ソリューションでの改善策

他の通知ソリューションによる改善策

- AlertManager
  - Prometheus の通知で使われているソフトウェア
  - ホスト名やトリガー ID など同じ属性の通知を集約して通知可能

Zabbix Active/Active の通知

- 複数 Zabbix からの同じ通知を
  一つの通知としてまとめて送ることも可能

全体のまとめ

- Zabbix のネイティブの機能をうまく使えば
  通知の効率化は可能

- シルバーバレットはないので、
  状況に応じて使い分けが必要

- 機能要望は Zabbix Feature Request へ
  - https://support.zabbix.com/projects/ZBXNEXT/

アンケートのお願い

https://questant.jp/q/5YTTTSWH

SRA OSS

www.sraoss.co.jp sales@sraoss.co.jp 03-5979-2701