
nic.br
Brazilian Network
Information Center
cgi.br
Brazilian Internet
Steering Committee
registro.br
cert.br
cetic.br
ceptro.br
ceweb.br
ix.br

Temporal
(Não, não é sobre chuva), mas também é sobre resiliência.

Como garantir resiliência em um ambiente de redes automatizado.

Adilson Torres <adilson@nic.br>

nic.br

Cronograma

- Sobre mim
- Automação “tradicional” e problema dos scripts frágeis
- Base fundamental de qualquer automação (Source of Truth)
- Temos uma SoT (Source of Truth) e agora?
- Temporal
    - Definição
    - Estrutura interna
    - Módulos e integrações
    - Arquitetura e fluxos
- Ex. 1 Exemplo básico
- Ex. 2 Padrão SAGA, rollback automático
- Ex. 3 Human-in-the-Loop - Aprovações no fluxo
- Ex. 4 Schedules - adeus CRON
- Ex. 5 SAGA + Human-in-the-Loop
- Temporal UI: Observabilidade e troubleshooting
- Boas práticas e próximos passos

Sobre mim

- Adilson Torres: Formado em Engenharia da Computação (Poli-USP), Engenheiro de Software e DevOps no IX.br
    - No NIC:
        - Contribui na equipe de ativação (2016-2018)
        - Contribui na equipe de desenvolvimento (2017-2019)
        - Desde 2023 no IX.br, trazendo novas tecnologias e melhorias (GraphQL, FastAPI, Temporal, Tailwind, HTMX, …)
    - Em outras empresas:
        - Incentivo a novos sistemas open source git (GITEA).
        - Implementação de portais (para clientes e equipes internas).
        - Automações em Ansible, Python, Bash.

Automação “tradicional”

- Scripts “Fire-and-forget” (executa e esquece).
  - Falhas durante execução, como resolve?
  - Tratamento de erros
  - Scripts, shell, cron e suas fragilidades
- Escalabilidade e manutenção
  - Quantidade de dispositivos para atuar e operar
  - Quantidade de scripts, crons, e outras automações para gerenciar e manter
  - Atualizações, mudanças e melhorias

Base fundamental de qualquer automação (Source of Truth)

- Mudança de fluxo:
  - Fluxo Tradicional: Engenheiro -> CLI -> Roteador (A rede é a fonte)
  - Fluxo NSoT: Engenheiro -> NSoT -> Automação -> Roteador (O código/banco de dados é a fonte)
- Fontes da verdade:
  - Netbox
  - Gitea (Servidor Git)
  - PostgreSQL (Banco de dados)

Temos uma SoT (Source of Truth) e agora?

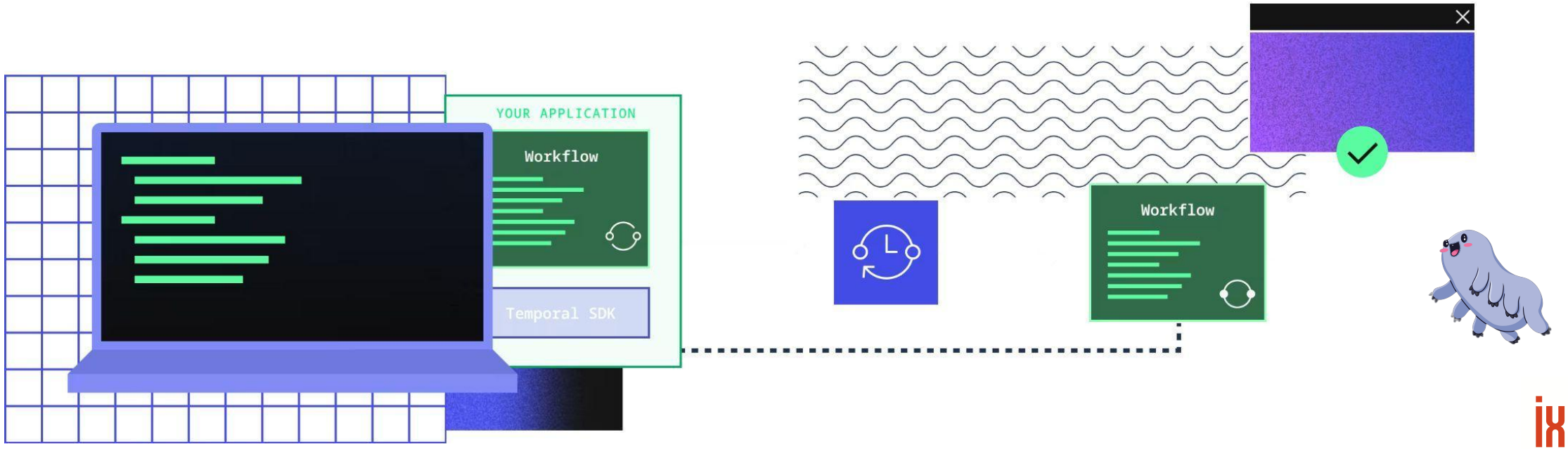
- Orquestração
  - Uma camada de controle
  - Conectar a fonte da verdade à execução, garantindo que no caminho nada se perca.
- Fazer os sistemas conversarem entre si
- Event-Driven Networking (EDN) e Intent-Based Networking (IBN)
- Um sistema que:
  - Plataforma de estado
  - Flexível (várias linguagens, modular, suporte a sistemas, bibliotecas existentes)
  - Tratamento de falhas integrado ou pelo menos fácil de implementar.

Temporal

Uma plataforma open-source criada para construir, gerenciar de uma maneira confiável e escalável aplicações tolerantes a falhas.

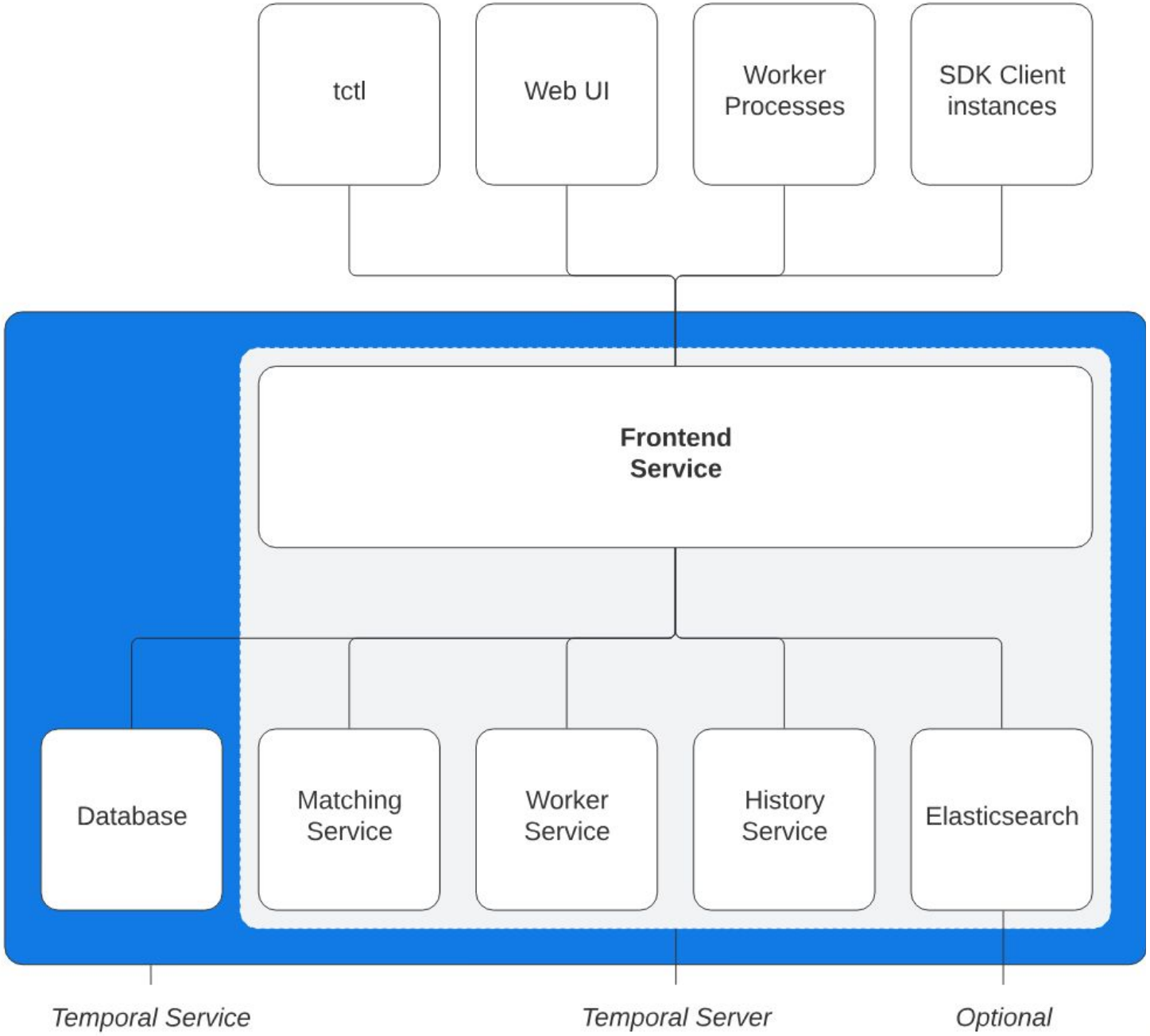
Orquestração de workflows via código, no qual o desenvolvimento é mais focado na lógica de negócio, deixando que a plataforma lide com o tratamento das situações adversas (falhas de rede, crashes em servidor, etc.).

É uma plataforma que garante a execução durável do código da sua aplicação.



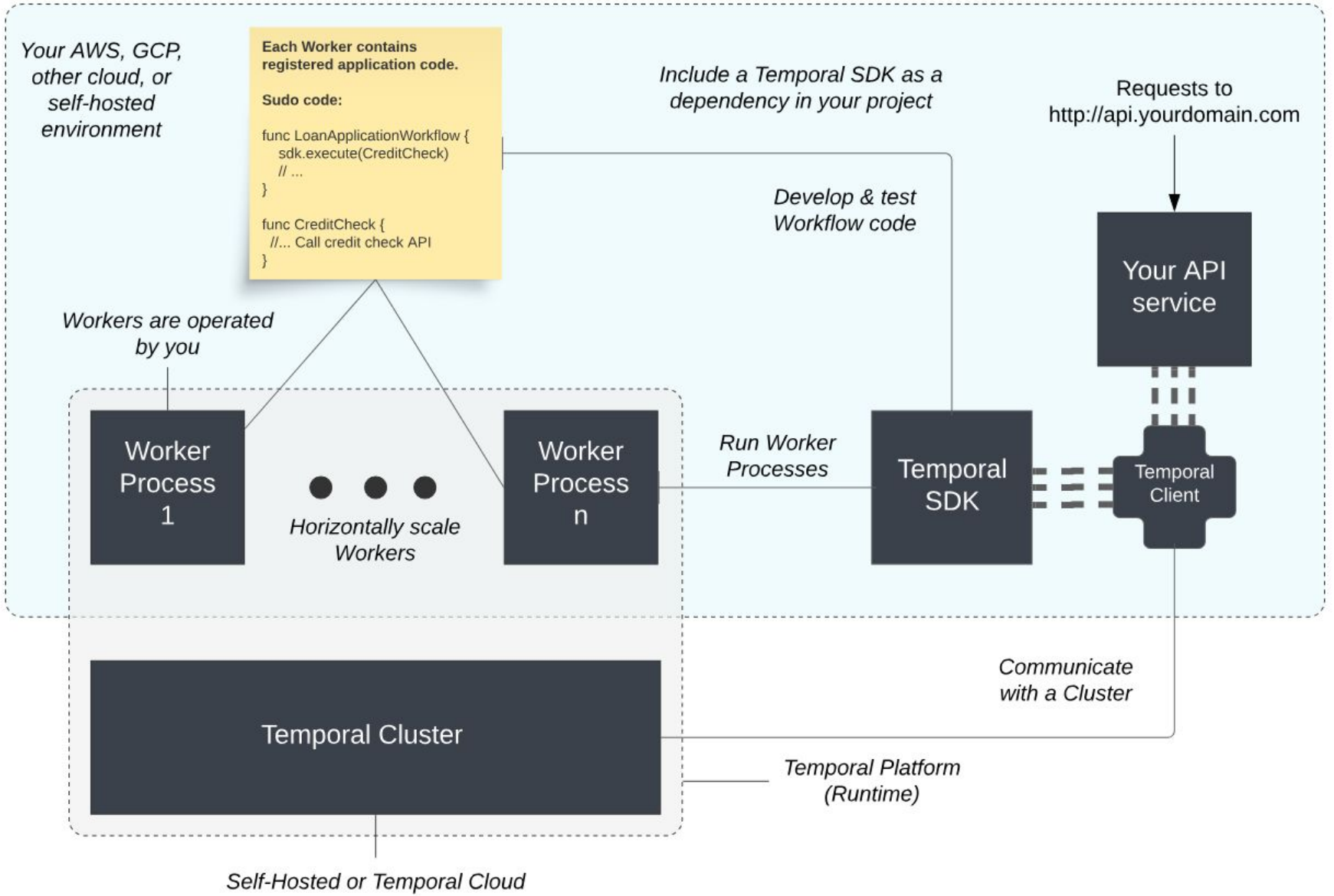
Temporal - Serviços

- Desenvolvido em Go
- Serviços internos utilizam gRPC para comunicação.
- Serviços internos:
  - History: Histórico de filas, dados, timers, eventos, execuções.
  - Matching: Roteamento das tarefas aos workers respectivos à fila.
  - Worker: Executa tarefas de background.
  - Frontend: Cuida do roteamento, rate limiting e autorização.
  - Visibility: Opcional



Temporal

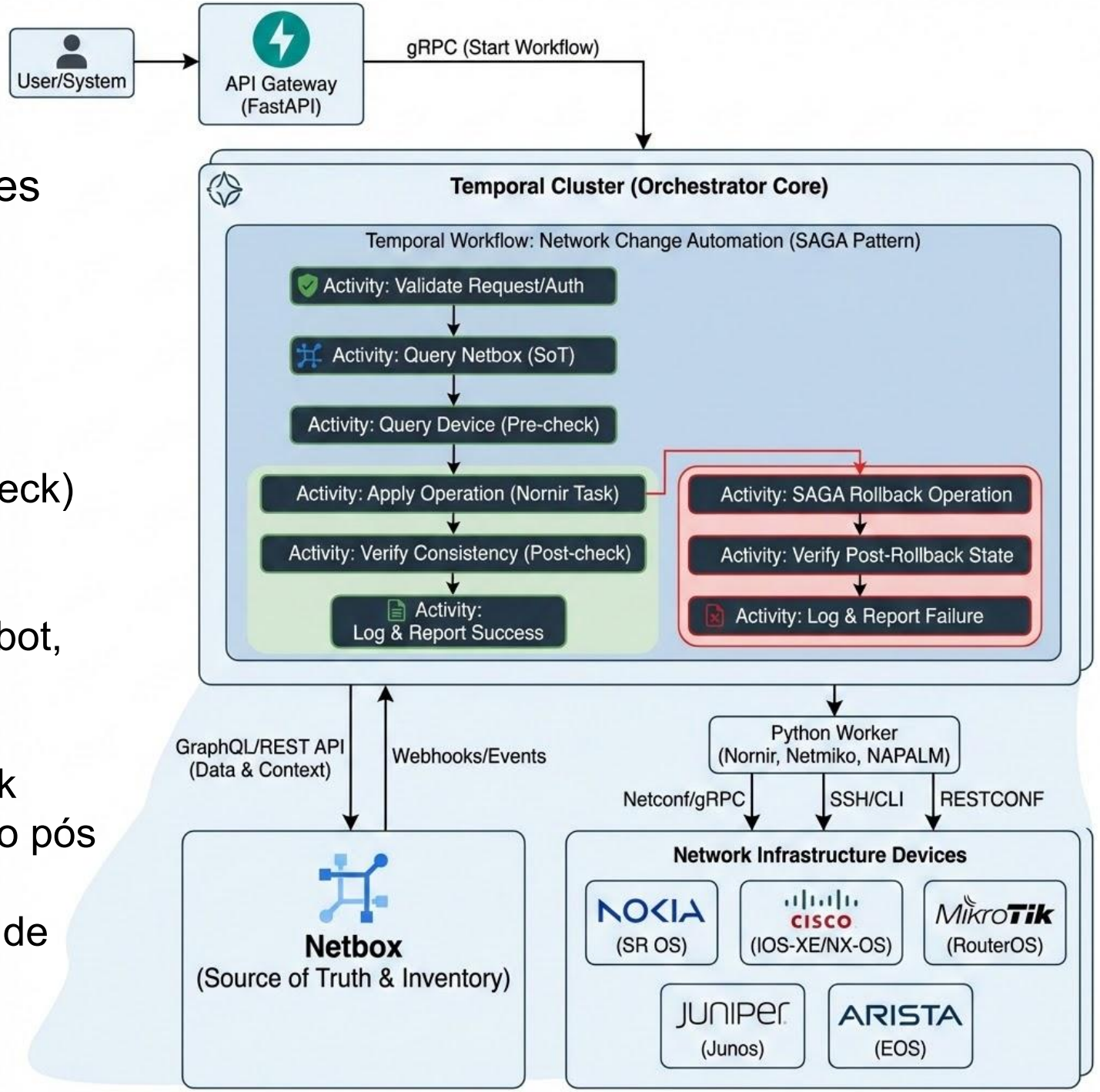
- Desenvolvimento com o uso de SDKs (Python, Go, PHP, ..)
- Além dos serviços internos do Temporal temos:
  - Client
  - Worker
- Código estruturado em:
  - Workflows
  - Activities



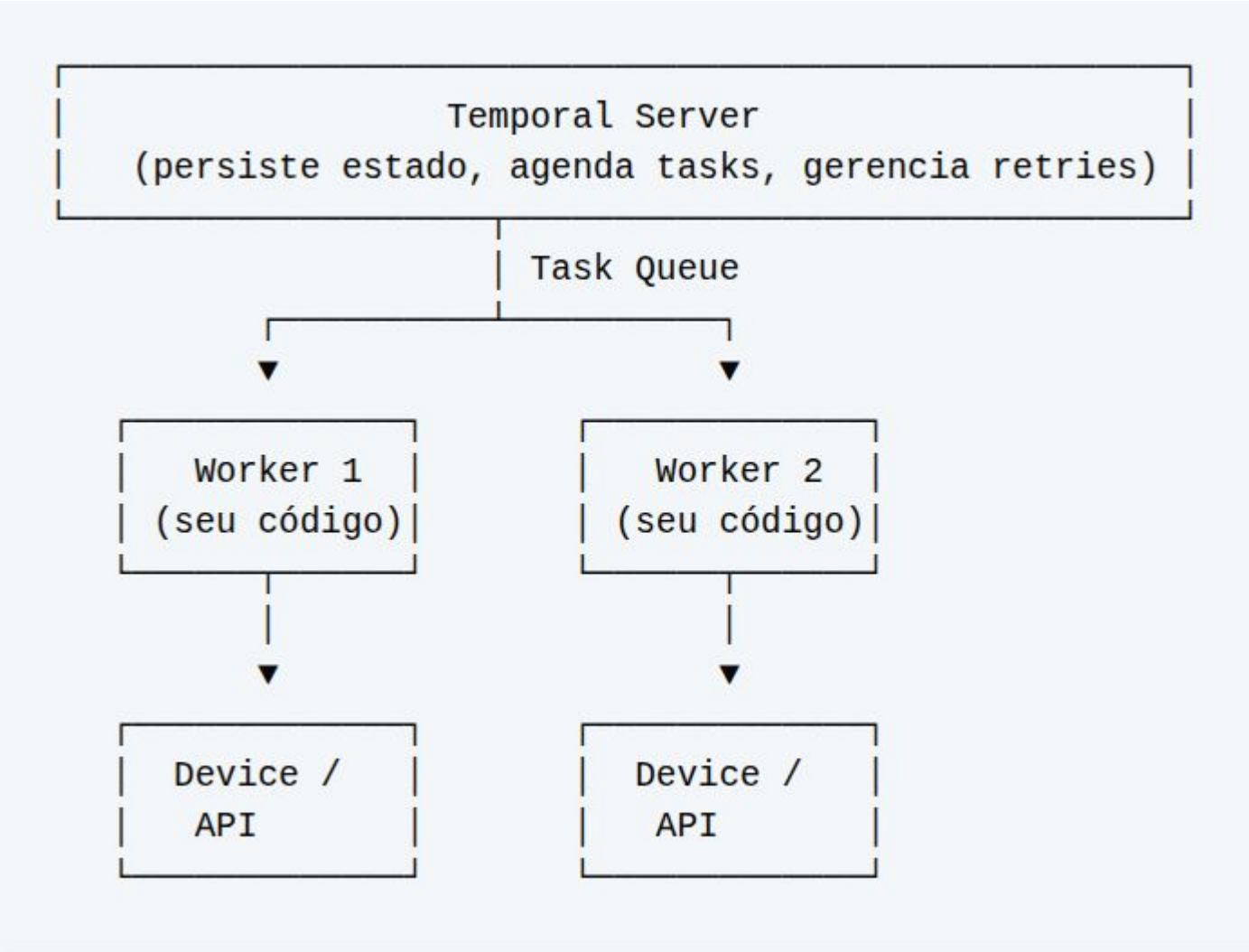
Temporal

Arquitetura e fluxo de ações

- Solicitação da atividade
- Validação da requisição e autenticação
- Consultas no Netbox
- Consultas no Device (pre-check)
- Aplica operação (Nornir)
- Verificação de consistência
- Logs, reporta sucesso (Chatbot, Slack, e-mail)
- Caso de falhas
    - Operação de rollback
    - Verificação do estado pós rollback
    - Logs e reporta falha de operação

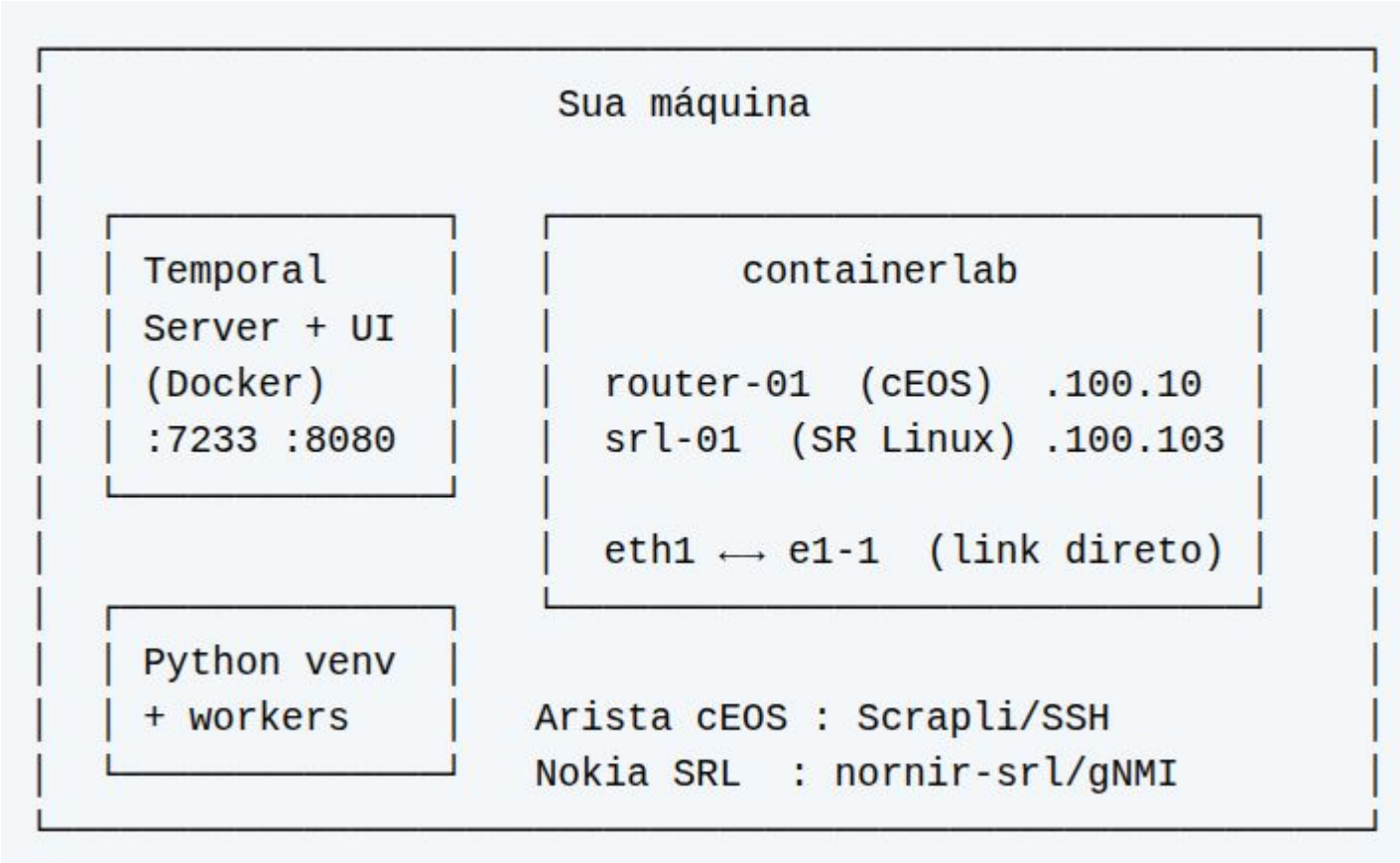


Temporal

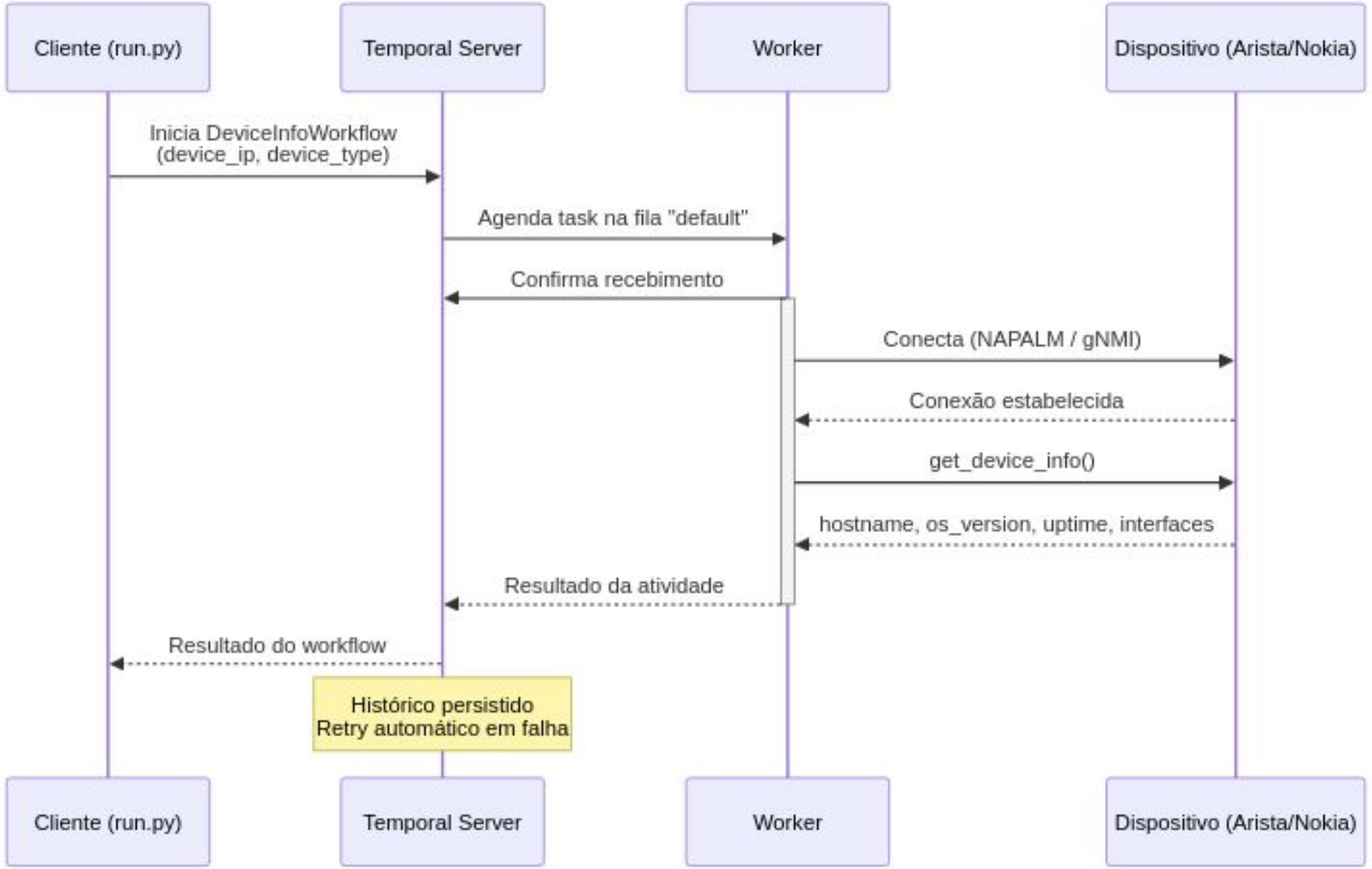

Temporal Server
(persiste estado, agenda tasks, gerencia retries)
Task Queue
Worker 1
(seu código)
Worker 2
(seu código)
Device /
API
Device /
API

Laboratório - Estrutura

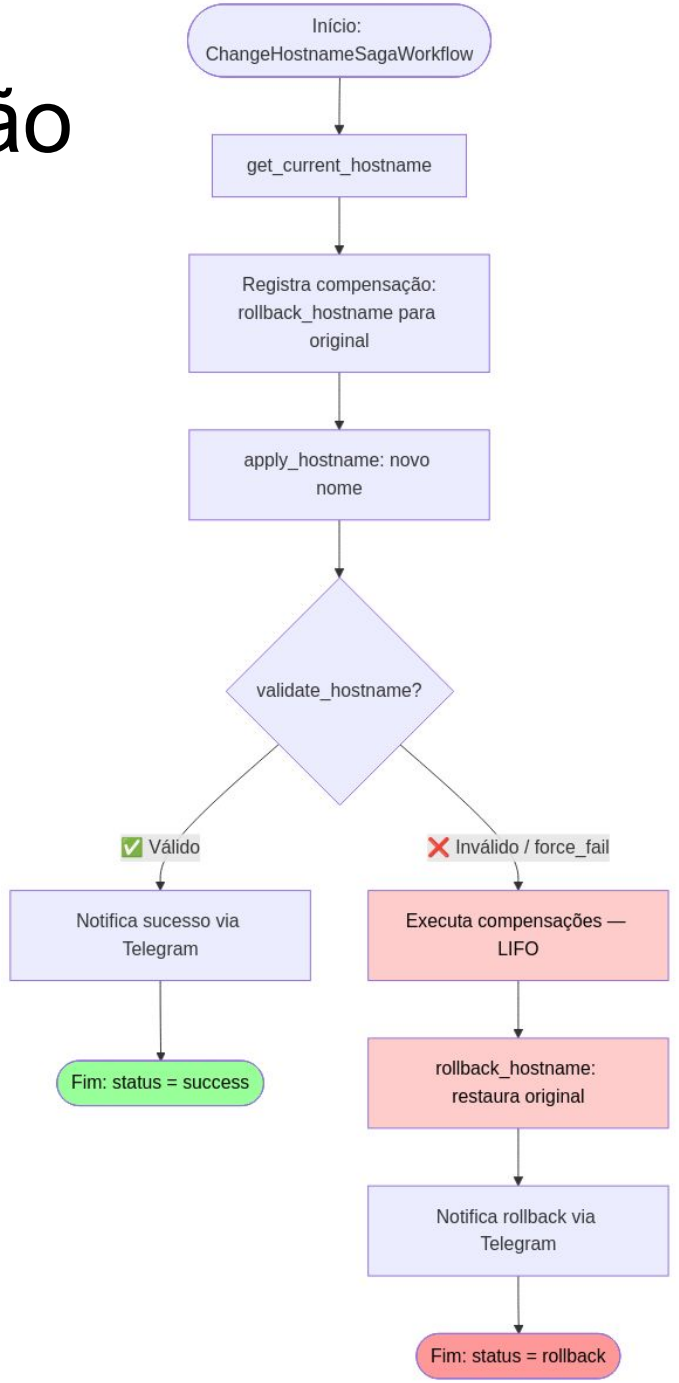
- Link do github
- Estrutura do lab:
  - Containerlab
  - Exercícios
  - Comandos
  - Links e referências



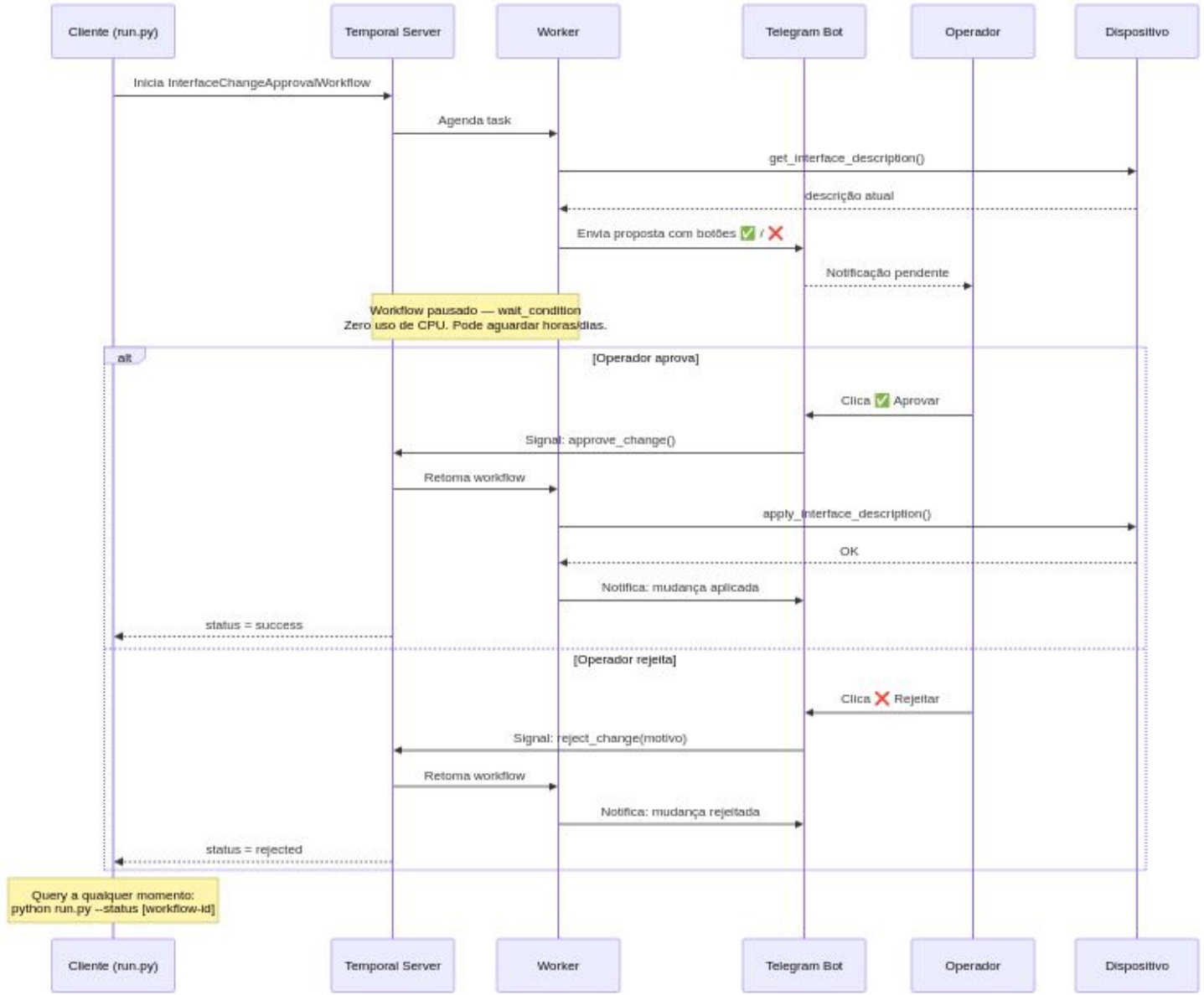
Exercício 01 - Workflow básico



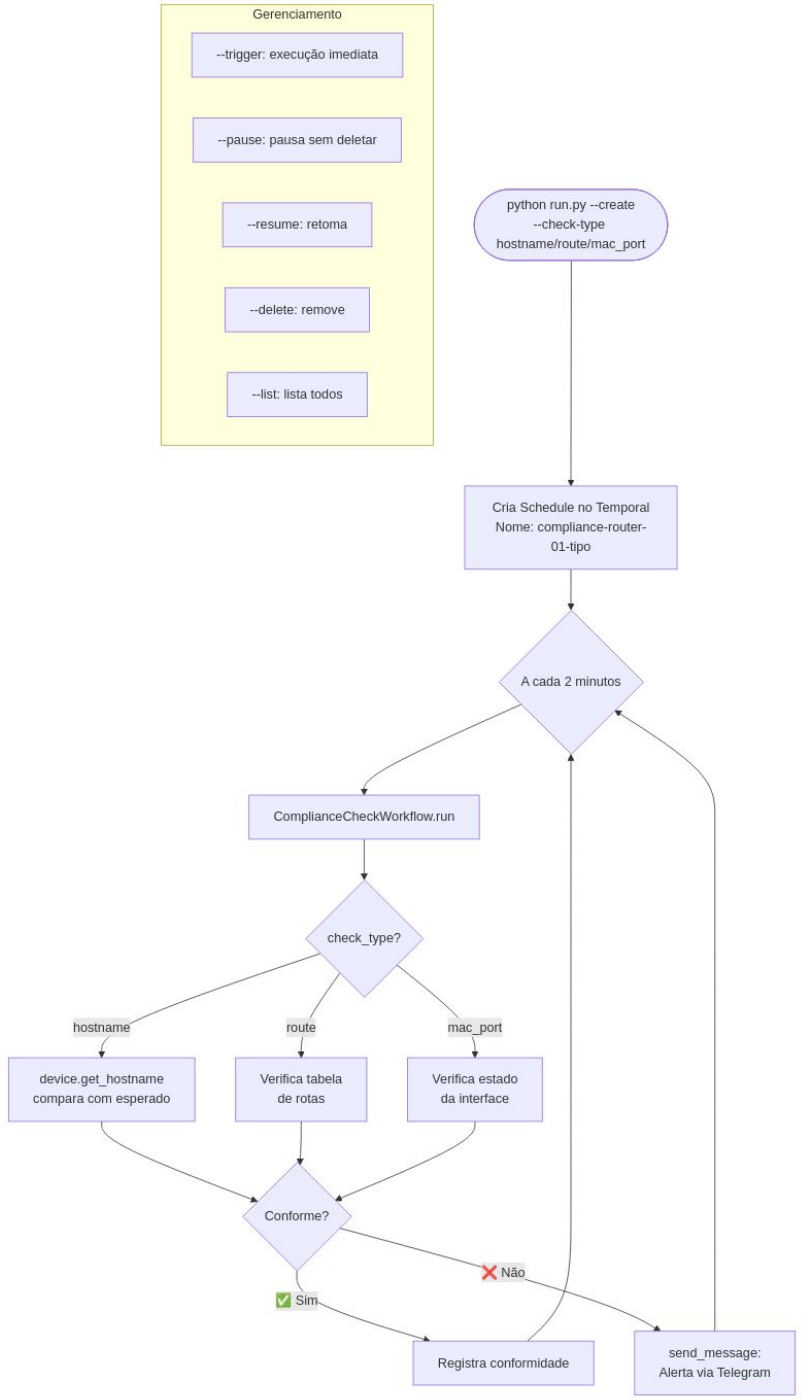
Exercício 02 - SAGA - Configuração com rollback automático



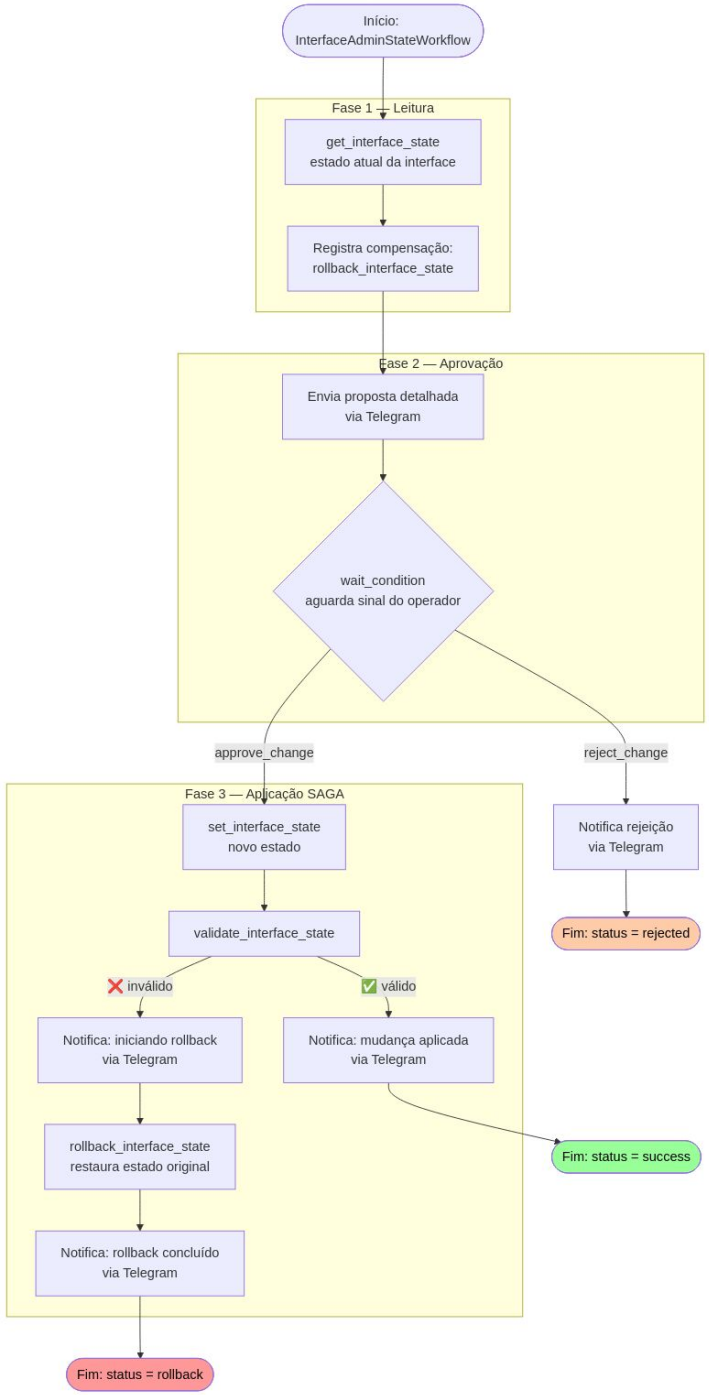
Exercício 03 - Human-in-the-Loop: Aprovação no fluxo



Exercício 04 - Schedules: Adeus CRON



Exercício 05 - SAGA + Human-in-the-Loop



Temporal UI - Observabilidade e Troubleshooting

- Timeline
- Input/Output
- Retry history
- Signal history
- Estado atual

Boas práticas e próximos passos

- Atividades devem ser idempotentes
  - Se ela rodar duas ou mais vezes com os mesmos inputs, o resultado deve ser o mesmo. (Falhas)
- I/O (Entradas e Saídas) em Workflows
  - Workflows devem ser determinísticos, acesso a recursos e outros sistemas devem estar dentro de Atividades.
- Valores padrão de policies:
  - maximum_attempts, initial_interval, backoff_coefficient
  - Timeouts
  - Pensar no comportamento esperado e definir os valores

Boas práticas e próximos passos

- Utilizar as filas (Task Queues) por domínio
  - device-queue, netbox-queue, notify-queue
  - Escalabilidade de workers
- Próximos passos:
  - Updates
  - Continue-as-New
  - Execução Paralela
  - Nexus
  - Versioning

Documentações

- Documentação: https://docs.temporal.io
- Python SDK: https://docs.temporal.io/develop/python
- Tutoriais: https://learn.temporal.io
- Comunidade: https://community.temporal.io
- Projeto desta apresentação: https://github.com/[usuario]/enfrentando-o-temporal
- Projeto anterior (event-driven-automation): referência de integração com NetBox + FastAPI

Obrigado

https://ix.br

Dúvidas: adilson@nic.br

Abril de 2026

nic.br cgi.br

www.nic.br | www.cgi.br