T4iSB

DOCUMENTO | **ESPECIFICAÇÃO TÉCNICA**

PRODUTO | **SISTEMA AUTOMATIZADO DE IDENTIFICAÇÃO BIOMÉTRICA (ABIS)**

FABRICANTE | **T4ISB TECNOLOGIA E PARTICIPAÇÕES LTDA.**

VERSÃO | **2022.1**

CONFIDENCIALIDADE

Este documento contém informações de propriedade da T4iSB BioLogica, sendo disponibilizadas de forma confidencial para uma finalidade específica. O destinatário garante a custódia e o controle, e concorda que este documento não será copiado ou reproduzido no todo ou em parte, nem seu conteúdo revelado de qualquer forma ou a qualquer pessoa, exceto no cumprimento da finalidade para a qual foi entregue.

Esta descrição se aplica a todas as páginas deste documento.

Este documento também poderá ser obtido a partir do endereço https://www.t4isb.com.

SUMÁRIO

1. INTRODUÇÃO

Este documento apresenta a arquitetura do ABIS da T4iSB BioLogica e seus componentes. Não é objetivo deste documento, apresentar dimensionamento de infraestrutura, já que os cenários de cada cliente poderão exigir dimensionamentos e/ou configurações diferentes, porém, o produto é flexível o bastante, para se adaptar às diversas situações de modelos computacionais (*On-primises* e/ou *Nuvem*).

2. CARACTERÍSTICAS GERAIS

- ABIS com capacidade de utilização em processos de unicidade biométrica de cadastros biométricos formados por até 10 impressões digitais e/ou fotografia facial cada.
- Capaz de realizar operações de identificação 1:N (um para muitos), comparando as novas biometrias inseridas na base de dados biométricos com as biometrias já disponíveis previamente na mesma base, garantindo que não exista duplicidade de biometria neste universo.
- Capaz de ser utilizados em processos que requeiram técnicas de eliminação de cópias duplicadas de dados repetidos em base, garantindo a individualização dos dados biométricos.
- Capaz de garantir a unicidade das identificações, impressões digitais e face, por meio de pesquisa biométrica de 1:N, com percentual de acerto acima de 99% (noventa e nove por cento).
- Capaz de produzir registro do resultado do processo de deduplicação da base com as informações das biometrias duplicadas.
- Capacidade de armazenamento flexível, sendo facilmente adaptável às necessidades do cliente, podendo armazenar cadastros biométricos compostos por até 10 impressões digitais e fotografia facial.
- Suporta concomitantemente sem interferência na qualidade e desempenho do seu funcionamento, biometrias de impressões digitais roladas e pousadas e biometria facial.
- Capaz de ser utilizado em processos de internalização de cadastros multibiométricos com até 10 impressões digitais e 1 fotografia da face.
- Capaz de ser utilizado em processos de deduplicação de cadastros multibiométricos com até 10 impressões digitais e 1 fotografia da face.
- Capaz de ser utilizado em processos de identificação 1:n de impressão digital, sobre uma base de dados de cadastros de impressões digitais deduplicados.

- Capaz de ser utilizado em processos de identificação 1:n de fotografia facial, sobre uma base de dados de cadastros de fotografias faciais deduplicados.
- Capaz de ser utilizado em processos de verificação 1:1 (um para um) de impressões digitais e do registro da face.
- Arquitetura robusta e facilmente escalável, capaz de manter tempo de resposta aderente às necessidades do cliente, mesmo em cenários de pico de transações simultâneas, aplicável em processos de internalização, deduplicação, identificação 1:n, verificação 1:1 e demais aplicações de uso do ABIS, a partir de template no padrão ISO.
- Suporte a imagens em formato WSQ para digitais, JPEG para face, ou templates, nos padrões ISO 19.794-2 e ISO 19.794-5, para impressões digitais e face, respectivamente.
- Comportamento agnóstico à origem da coleta da imagem, no recebimento de imagem em formato JPEG para verificação.
- Capaz de disponibilizar recursos para análise e aprovação humana das imagens WSQ e JPEG submetidas para verificação.
- Capaz de realizar a verificação entre dois padrões biométricos submetidos simultaneamente, independentemente de estarem previamente cadastrados no ABIS, sendo suportados os padrões JPEG, WSQ e templates nos padrões ISO 19.794-2 e ISO 19.794-5.

Capaz de realizar a verificação entre um padrão biométrico submetido pela aplicação cliente e outro de mesmo tipo previamente cadastrado no ABIS, sendo suportados os padrões JPEG, WSQ e templates nos padrões ISO 19.794-2 e ISO 19.794-5.

- Suporte ao padrão ICAO na verificação de imagens.
- Capaz de ser integrado por meio de API, com outros sistemas que requeiram operações biométricas (ex.: sistema de cadastramento biométrico, autenticação biométrica...).

A arquitetura de microsserviços do T4-AMEN permite de forma escalável e transparente, o uso de diversas tecnologias de extração e comparação de templates de forma concomitante. Essa inovadora arquitetura permite que, dependendo da situação e da qualidade das informações, diferentes provedores sejam utilizados no processamento biométrico.

Para atingir performance de classe Mundial nessas operações, a versão atual do ABIS T4ISB utiliza tecnologia MIAXIS e BIOLÓGICA para impressões digitais e T4ISB para facial, todas tecnologias certificadas e listadas no mais alto grau de rigor pelo NIST.

A comparação de templates de impressão digital é feita pela tecnologia MIAXIS e está listada na página https://pages.nist.gov/minex/results/tables/ do NIST, certificando alto nível de acurácia.

Já a extração de templates, faz uso tanto do extrator MIAXIS como do extrator BIOLÓGICA. Este por sua vez é o extrator mais rápido do mundo na listagem oferecida pelo NIST, proporcionando altíssima velocidade de extração em negócios de missão crítica com baixa necessidade de infraestrutura computacional.

Miaxis Biometrics Co. Ltd.	Native Single Finger FNMR @ FMR = 0.0001	Native Two Finger FNMR @ FMR = 0.01
	0.0149	0.00035

3. VISÃO GERAL DO PRODUTO

Ao longo do documento, o ABIS será referenciado por T4-AMEN (T4ISB *Automated Matching ENgine*), que é um produto pertencente à solução T4ISB IIMP (T4ISB *Integrated Identity Management Platform*) apresentada na ilustração a seguir:

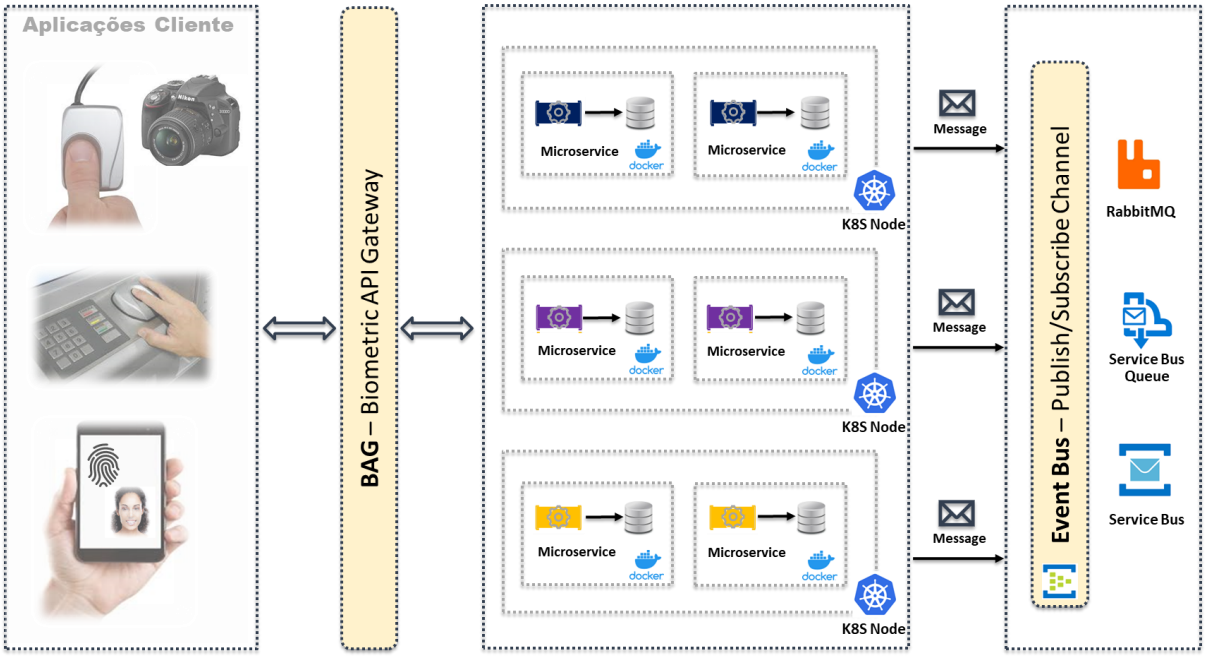

Figura 1 – Visão Geral da Arquitetura T4ISB IIMP (T4ISB Integrated Identity Management Platform)

I. **APLICAÇÕES CLIENTE:** Softwares utilizados para captura de impressões digitais e/ou faciais, por meio de estações de trabalho conectadas a dispositivos de coleta de impressões digitais e câmeras fotográficas ou coleta da imagem das impressões digitais ou face, a partir de dispositivos móveis.

II. **BAG (*Biometric API Gateway)*:** Responsável pela governança da identidade, gerenciamento de certificados, roteamento das requisições e gerenciamento do fluxo de trabalho.

III. **SERVIÇOS:** A arquitetura do produto é formada por serviços com atribuições específicas, possibilitando que o crescimento do ABIS ocorra conforme necessidade específica, evitando a ociosidade de recursos que não necessitam de novas instâncias. Os serviços são

hospedados em contêineres utilizados para virtualização, fazendo uso do Kubernets como sistema de orquestração, automatizando assim, a implantação, o dimensionamento e a gestão de aplicações hospedadas em contêineres.

Microsserviços

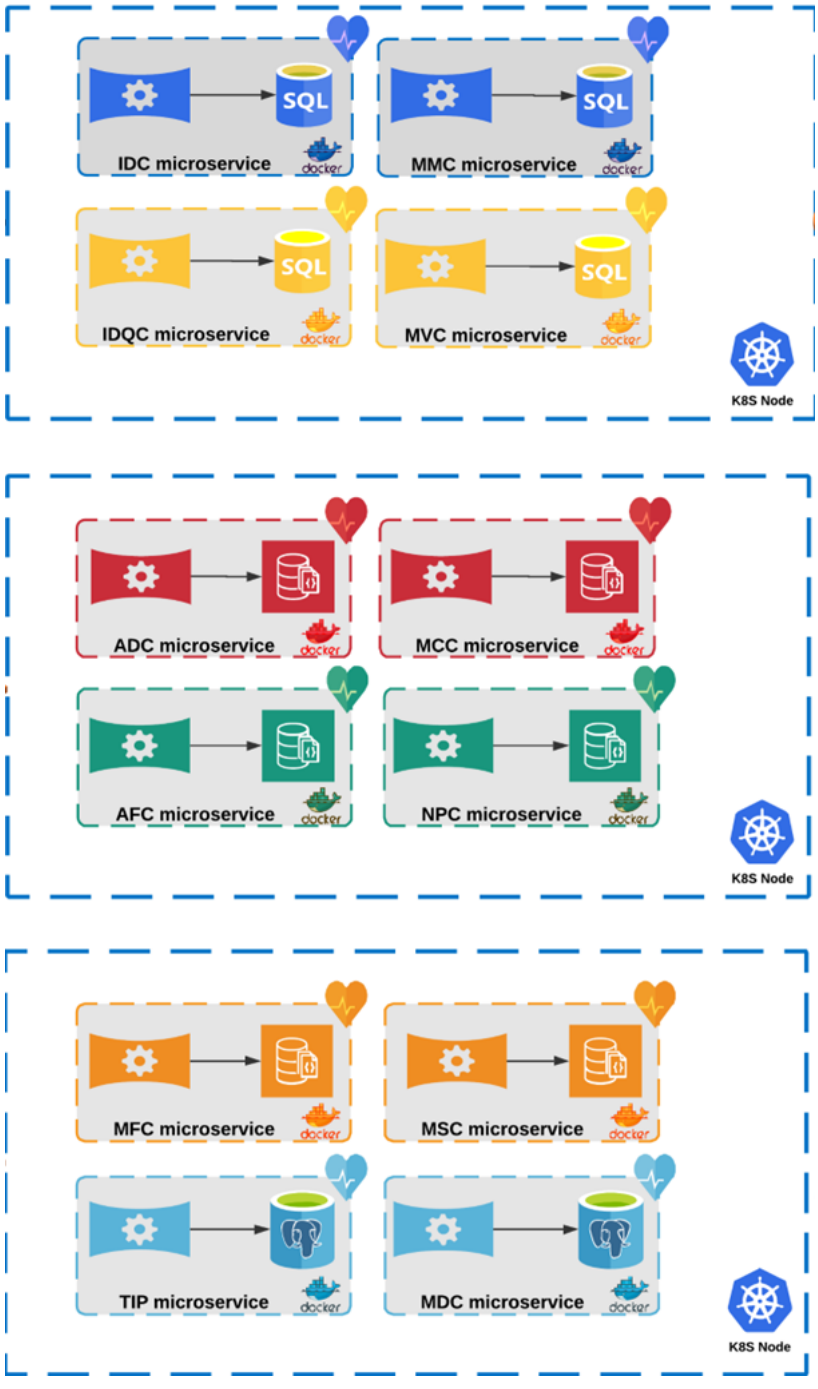

Figura 2 – Microsserviços

- **IDC (*Identity Controller*):** Responsável pela Verificação e Autorização de Identidade
- **IDQC (*ID Query Controller*):** Responsável por consultar o banco de dados para verificar o ID do registro
- **MMC (*Micro Matching Controller*):** Responsável por requisições e transações de identificação 1:N
- **MVC (*Micro Verification Controller*):** Responsável por requisições e transações de verificação 1:1
- **AFC (*Arquive Frontend Controller*):** Responsável por consultar as bases de dados do arquivo de registros.
- **MCC (*Micro Coding Controller*):** Responsável por todas operações e transações de codificação.
- **NPC (*Nist Proxy Controller*):** Responsável por todas operações e conversões de arquivos NIST.
- **MFC (*Main Frontend Controller*):** Responsável pela comunicação com o principal *frontend* SPA (Aplicações de Página Única).
- **TIP (*Transaction in Progress*):** Responsável pela gestão das transações em andamento.
- **MSC (*Micro Sequence Controller*):** Responsável por todas as transações e operações de verificação de sequência.
- **MDC (*Micro Directory Controller*):** Responsável pelas operações de diretório.
- **MEC (*Micro External interfaces Controller*):** Responsável pela gestão dos sistemas externos.
- **MNC (Micro Nist Controller Service):** Responsável pelas operações do hub biométrico utilizando o padrão Nist.
- **MZCS (Micro Zeebe Controller Service):** Responsável pela comunicação do motor BPMN utilizando protocolos Grpc.
- **Nginx-reverse-proxy:** Usado como proxy reverso para todos os serviços acima.
- **Nginx-frontend:** usado como servidor de aplicativos para o IIMP Frontend Web Application.
- **Zeebe:** Motor Camunda BPMN.

IV. **EVENT BUS:** Responsável pela gestão de eventos e mensagens, incluindo o controle de fila

4. ARQUITETURA T4-AMEN

Nós Lógicos

O cluster de correspondência do T4-AMEN é formado por um número de nós idênticos, sendo totalmente simétrico e não existindo componentes centrais. Cada servidor no cluster executa apenas uma instância do software de cluster correspondente, mas esse único processo integra internamente três tipos de nós lógicos:

- ***API node*:** Expõe a API REST do cluster correspondente. Os clientes podem se conectar a qualquer nó ativo no cluster. O *API node* executa solicitações leves diretamente. Solicitações pesadas são encaminhadas de forma aleatória para o *Worker node*.
- ***Worker node*:** Lida com solicitações pesadas, principalmente extração de recursos e combinação de resultados de identificação retornados por *Index nodes* individuais.
- ***Index node:*** Armazena em cache todos os *templates* na memória. Cada *Index node* recebe um compartilhamento do banco de dados para armazenar em cache. Cada *Index node* realiza a verificação 1:1 e identificação 1:N em pessoas existentes em seu cache.

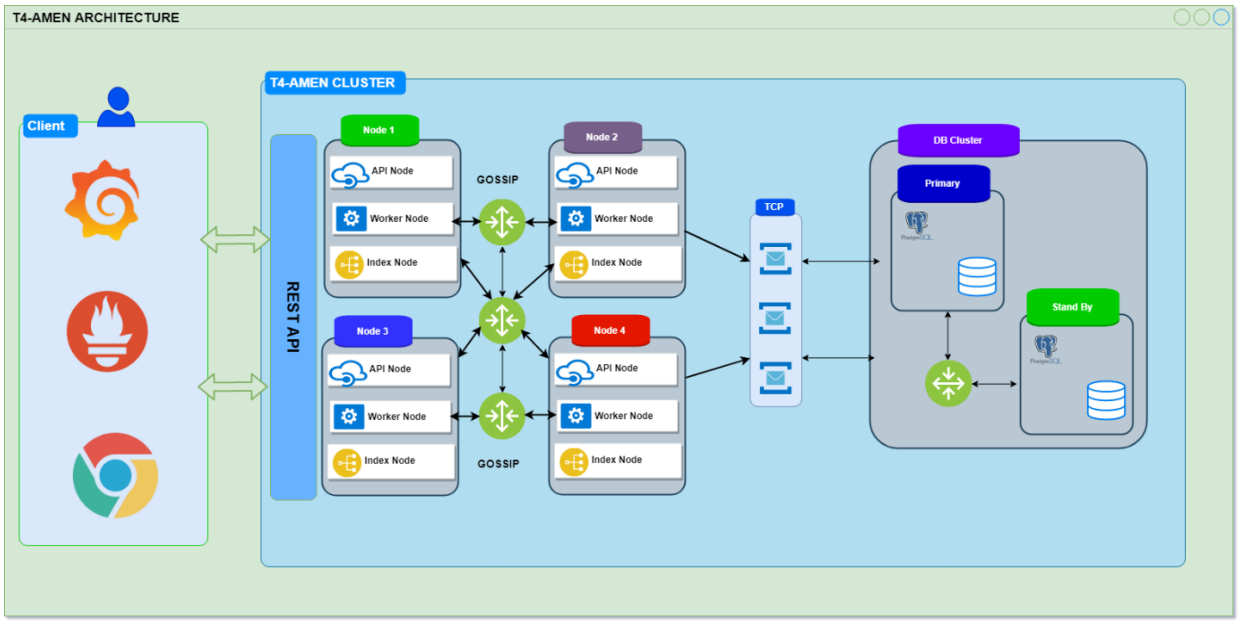

Figura 3 – Nós lógicos – T4-AMEN

O *Worker node* e *Index node* expõem sua funcionalidade em uma porta separada que é usada apenas para comunicação entre nós (também chamada de "gossip"), para que diferentes medidas de segurança possam ser aplicadas às portas de API e de "gossip".

Desempenho

O cluster distribui operações pesadas (principalmente extrações e identificações de recursos) uniformemente em todos os nós, independentemente de qual nó é usado pelos clientes para emitir solicitações. Se for observado que algum nó está sobrecarregado ou subutilizado durante a operação do cluster, a carga do nó pode ser ajustada conforme necessidade.

A paralelização automática do cluster é suficiente para utilizar mais de 90% da capacidade de hardware para carga de trabalho de identificação. O cluster irá canalizar trabalhos de identificação consecutivos sem pausas desnecessárias se pelo menos dois clientes estiverem solicitando identificações simultaneamente. Algumas operações são inerentemente sequenciais (extração de recursos de impressão digital única, verificação única e até mesmo algumas partes da identificação).

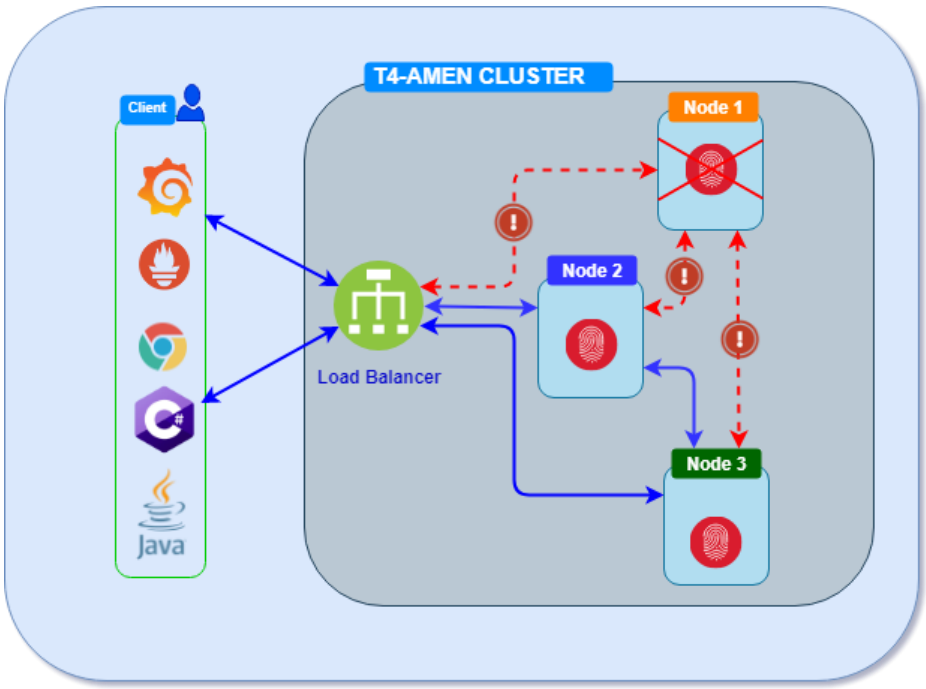

Figura 4 – Desempenho – T4-AMEN

Quando o cluster estiver sobrecarregado, ele enfileirará solicitações e as executará na ordem em que foram recebidas. Os clientes não devem cancelar e repetir operações lentas, pois isso resultaria em uma avalanche de solicitações que aumentaria ainda mais a carga no cluster já sobrecarregado. Clientes (ou threads de cliente em clientes multithread) devem sempre esperar a última operação ser concluída (se importando com o resultado ou não) e só então emitir a próxima solicitação. Para garantir esse comportamento, os clientes devem ser configurados com tempos limite muito longos (até uma hora). Interfaces de usuários interativas devem emitir solicitações por meio de thread em segundo plano.

Como os nós não armazenam *templates* em cache no disco, a reinicialização do nó faz com que ele busque seus intervalos de pessoas atribuídos no banco de dados. Quando o cluster é reiniciado, todos os nós executam essa busca simultaneamente. Supondo que o banco de dados armazene dados em SSDs ou em armazenamento com desempenho equivalente, a busca de *template* deve saturar o link de 1 Gbps para o banco de dados, o que se traduz em cerca de 200.000 *templates* por segundo. Quando apenas um pequeno nó é reiniciado, o carregamento do *template* pode ser limitado à CPU em vez de limitado à rede. Como o SSD ou equivalente pode ser muito caro para armazenamento de imagens, as imagens são armazenadas em uma tabela separada, que pode ser localizada fisicamente em discos rígidos mais baratos.

O peso do nó determina a quantidade relativa de carga de trabalho atribuída a nós individuais, mas o peso médio do nó também influencia o comportamento geral do cluster. Um peso médio de nó mais alto resulta em uma distribuição de carga mais uniforme ao custo de alguma sobrecarga computacional incorrida por cada solicitação. O peso de nó padrão de 1.000 oferece uma compensação aceitável.

Resiliência

Os nós fazem ping uns aos outros a cada segundo. Se algum nó cair, outros nós perceberão isso em segundos, porque os pings falharão ou travarão. Uma vez que o nó é reconhecido como indisponível, deixarão de enviar solicitações para ele. Se o nó ficar pronto novamente, outros nós perceberão isso em segundos e começarão a enviar solicitações para esse nó novamente.

O cluster mede dois tipos de disponibilidade:

- **Capacidade**: Quantidade de poder computacional disponível. É definido como contagem ponderada de nós prontos. Por exemplo, se houver 10 nós e 3 estiverem inativos, a capacidade será de 70%.
- **Cobertura**: Definida como a probabilidade de que qualquer pessoa seja armazenada em cache por pelo menos um nó.

O cluster é configurado com um fator de replicação. O fator de replicação de 3 significa que cada pessoa é armazenada em cache em três nós diferentes. Se o fator de replicação for 1, a capacidade e a cobertura serão as mesmas. Se o fator de replicação for 2 ou mais, a cobertura será muito maior que a capacidade. Se o número de nós indisponíveis for menor que o fator de replicação (por exemplo, 2 nós inativos com fator de replicação 3), a cobertura de pessoas permanecerá 100%. Mas mesmo com interrupções maiores, a cobertura permanece alta. Por exemplo, com 10 nós e fator de replicação de 2, a perda de 2 nós resultará em 80% de capacidade e 97-98% de cobertura. Continuando este exemplo, com 3 nós perdidos, a capacidade cai para 70% enquanto a cobertura cai para 93-94%.

O cluster de correspondência T4-AMEN continua a atender solicitações enquanto pelo menos um nó estiver ativo. Mesmo com cobertura abaixo de 100%, verificações não são afetadas e todas as identificações podem continuar. Os clientes podem especificar a cobertura mínima para cada solicitação de identificação.

Fator de replicação	Nós indisponíveis	Cobertura	Capacitade
1	0 a 1	100%	100%
3	2 a 3	100%	33%
2	2 a 10	97-98%	80%
2	3 a 10	93-94%	70%

Tabela 1 – Exemplos de Resiliência T4-AMEN

Consistência

O armazenamento em cache de *templates* na RAM gera risco de inconsistências com o banco de dados. A replicação de pessoas em vários nós, gera risco de inconsistências entre os nós. Vários clientes simultâneos podem gerar um risco de gravações conflitantes.

Contudo, o cluster de correspondência T4-AMEN emprega várias estratégias para evitar estes riscos e garantir a consistência.

É comum que dois clientes tentem modificar a mesma pessoa simultaneamente (um deles será o primeiro). A interrupção temporária da rede resulta apenas em inconsistências temporárias. Os nós serão sincronizados em segundos após a restauração da conectividade. Com controle adicional, os nós realizam uma nova varredura periódica do banco de dados, que define o limite superior no tempo de vida das inconsistências de cache.

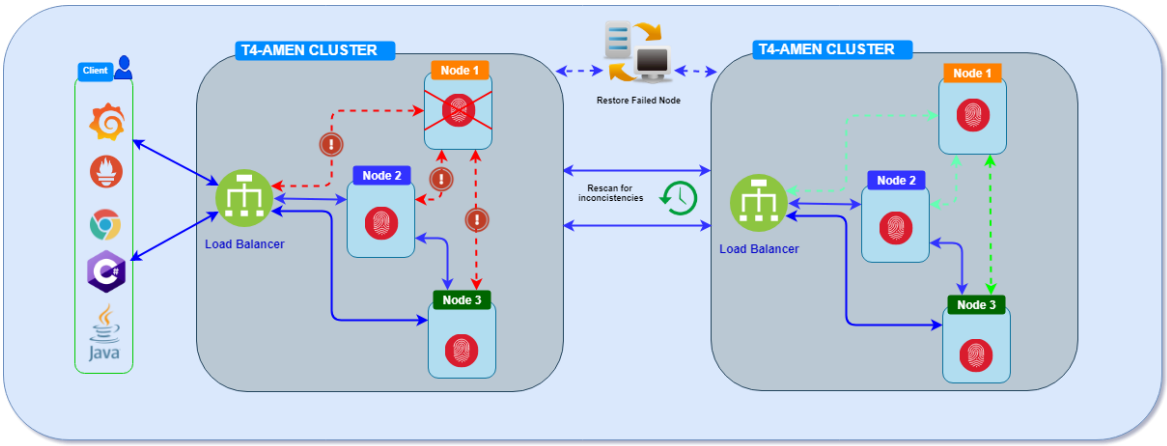

Figura 5 – Consistência T4-AMEN

Componentes

O cluster é composto por muitos componentes, que são alocados conforme o dimensionamento da infraestrutura, sendo os principais, descritos abaixo:

- **Apiserver**: Expõe a API do *Kubernetes*. É o front-end para o painel de controle do *Kubernetes*. Ele foi projetado para ser dimensionado horizontalmente, ou seja, consegue disponibilizar mais instâncias.
- **Etcd**: Todos os dados do cluster são armazenados aqui. Sendo essencial que sempre exista um plano de backup para os dados do "etcd" para seu cluster *Kubernetes*.
- **Scheduler**: Observa os pods recém-criados que não possuem nenhum nó atribuído e seleciona um nó para que sejam executados.
- **Controller Manager:** Executa controladores, que são os threads em segundo plano que tratam de tarefas de rotina no cluster. Logicamente, cada controlador é um processo separado, mas para reduzir a complexidade, todos eles são compilados em um único binário e executados em um único processo
- **Kubelet**: É o agente do nó primário. Ele observa os pods que foram atribuídos ao seu nó (por apiserver ou por meio do arquivo de configuração local)
- **Proxy**: Habilita a abstração do serviço Kubernetes mantendo as regras de rede no host e realizando o encaminhamento de conexão.
- **Load Balancer**: É composto por 2 serviços:
  - ✓ **Keepalived cluster:** Configura um endereço IP virtual (ex.: 192.168.20.10), esse endereço IP virtual aponta para os nós mestres do cluster.
  - ✓ **Nginx service:** Atua como o balanceador de carga do *apiserver* mestre. Os outros nós do *Worker* se conectam ao endereço IP virtual *keepalived* (192.168.20.10) e o *nginx* expõe a porta (16443) para se comunicar com os *apiserver*s do cluster mestre.

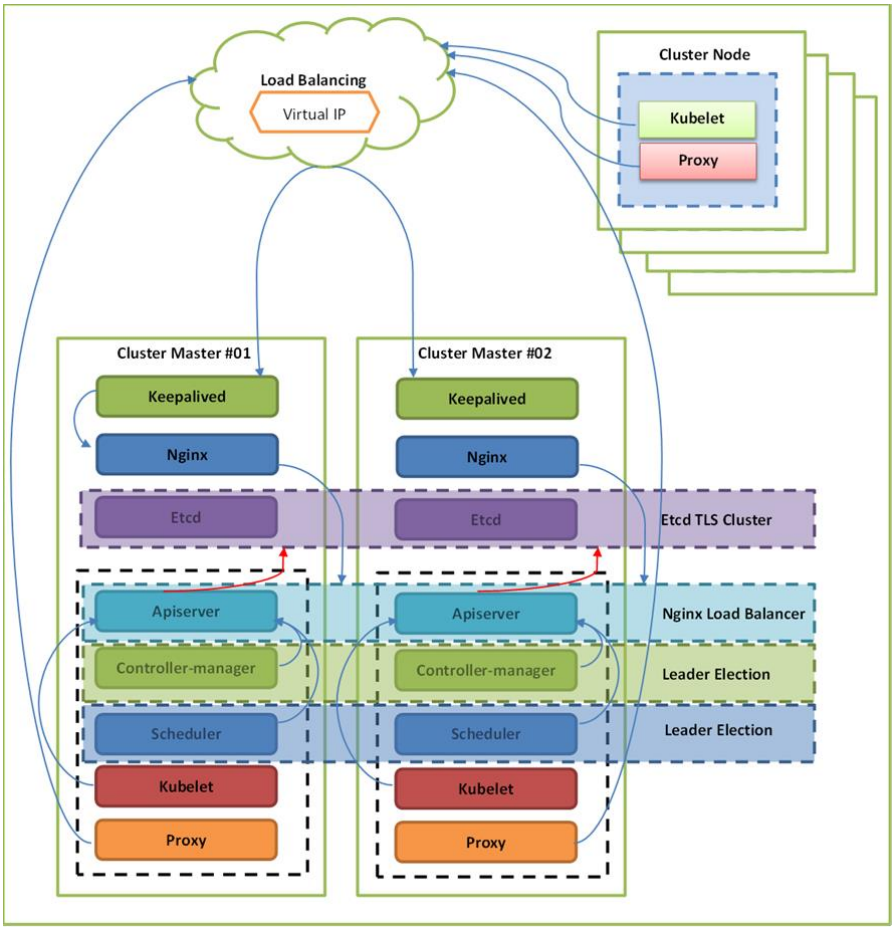

Figura 6 – Componentes do cluster

APIS

Cada nó no cluster expõe a seguinte API REST:

- **PUT /persons/{personId}** - Adiciona nova pessoa. Se a pessoa já existir, ela será substituída. O corpo da solicitação contém um objeto de pessoa codificado em CBOR *(Concise Binary Object Representation).*

- **DELETE /persons/{personId}** - Remove a pessoa. Se a pessoa não existir, este método não tem efeito.

- **GET /persons/{personId}** - Recupera a pessoa. A resposta 404 vazia indica que a pessoa não existe. O corpo da resposta contém um objeto de pessoa codificado por CBOR.

- **POST /persons/{personId}/audit** - Verifica a consistência da pessoa, recalcula dados redundantes (modelos) e sincroniza a pessoa nos caches do nó. Se a pessoa não existir,

execute apenas a sincronização do cache. Retorna uma resposta de erro (4xx ou 5xx) se os dados pessoais armazenados no banco de dados estiverem corrompidos.

- **POST /persons/{personId}/verify** - Executa a verificação 1:1 da pessoa. Se a pessoa com o ID fornecido não existir, uma resposta 404 vazia será retornada. A solicitação contém a pessoa examinada codificada por CBOR. A resposta contém uma pontuação de ponto flutuante codificada por CBOR que descreve a semelhança entre a pessoa examinada e as pessoas candidatas (cumulativa para todas as impressões digitais).

- **POST /identify** - Executa a identificação 1:N em todas as pessoas. A solicitação contém uma consulta de identificação codificada em CBOR. A resposta contém uma lista codificada por CBOR de objetos de correspondência

- **GET /live** - Sempre retorna uma resposta 200 vazia.

- **GET /ready** - Retorna uma resposta 200 vazia se o nó estiver pronto e uma resposta 503 se não estiver pronto. O nó está pronto quando seu cache foi carregado do banco de dados.

- **GET /cluster-ready?capacity={minCapacity}&coverage={minCoverage}`** - Retorna uma resposta 200 vazia se todo o cluster estiver pronto e uma resposta 503 se não estiver pronto. O controle é definido por dois parâmetros de ponto flutuante, ambos no intervalo de 0 a 1, sendo o padrão o valor 1. O cluster é considerado pronto se sua capacidade for pelo menos "minCapacity" e sua cobertura for pelo menos " minCobertura".

- **GET /status** - Retorna o objeto de relatório de status do cluster codificado por CBOR.

- **GET /prometheus** - Prometheus metrics scraping location.

Todas as APIs REST retornam o código HTTP 400 para erros permanentes (geralmente causados por parâmetros de solicitação inválidos) e 503 para erros temporários

A resposta de erro contém um objeto de erro codificado em CBOR. O cluster de correspondência não impõe nenhuma política de repetição para erros temporários. Ele apenas propagará erros temporários para clientes com respostas 503.

Monitoramento

O monitoramento está disponível por meio de métricas exportadas do *Prometheus*. O *endpoint* "GET/prometheus" está disponível para obter todas as métricas.

O T4-AMEN fornece um painel *Grafana* com algumas métricas úteis, no entanto, o *endpoint Prometheus* pode ser usado para adicionar novos painéis.

Figura 7 – Painel Grafana

5. INTEROPERABILIDADE

Nenhum dos serviços relacionados com o Sistema Automatizado de Identificação Biométrica (ABIS), depende de hardware ou de firmware específico de determinado fabricante.

Possui compatibilidade com templates ISO 19794-2 – Biometric data interchange formats – Finger minutiae data e com ANSI 378 – Finger Minutiae Format for Data Interchange.

O extrator de templates e o matcher descritos no início do documento, foram testados e aprovados pelo procedimento MINEX III do NIST, podendo ser comprovados a partir do endereço https://pages.nist.gov/minex/results/tables/.

Possui também, capacidade de extração e matching de templates compatíveis com ISO 19794-5 - Biometric data interchange formats – Face image data, estando o algoritmo, publicado pelo NIST/FRVT (Face Recognition Vendor Test) em sua última versão.

6. CAPACIDADE

A solução ABIS fornecida pela T4ISB pode ser instalada em ambiente *On-premises* ou Nuvem, ficando à escolha do cliente, a alternativa mais aderente às suas necessidades e possibilidade de crescimento rápido.

As licenças necessárias para operacionalização da solução são permanentes, de uso ilimitado e mantidas no cliente conforme suas condições contratuais.

A solução não possui restrição de quantidade de usuários cadastrados, quantidade de registros, nem de acessos simultâneos.

As interfaces existentes são entregues no idioma definido pelo cliente, assim como os manuais e documentações geradas.

Compatibilidade com os seguintes padrões:

- Debian 8 ou superiores;
- Windows 10 ou superiores;
- Periféricos – comunicação no padrão JXFS;
- Java version "1.8.0_131" 64-Bit;
- Padrão Oauth 2.0 e suas evoluções para se comunicar com sistema de permissionamento de usuários;
- Padrão JSON para comunicação com API de sistemas corporativos da instituição financeira para tráfego de informações biométricas.