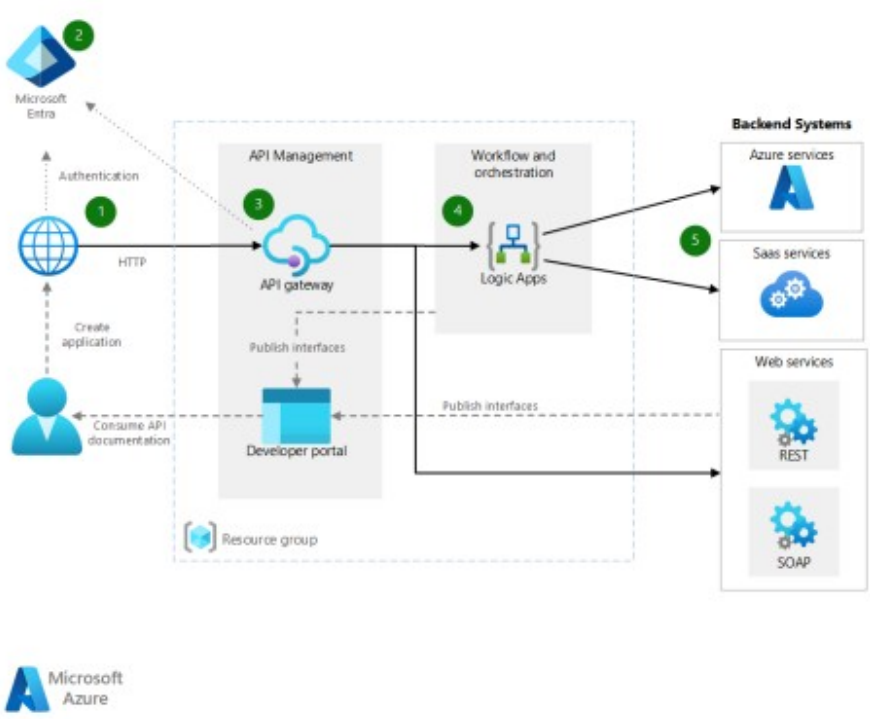
Relatório de análise da solução: tmpqimna7rn



Análise completa da solução atual

Modelo de cloud:
- Microsoft Azure

Lista com os componentes:
- Consumidor/cliente de API (aplicação web/mobile/sistema terceiro)
- Microsoft Entra ID (antigo Azure AD) para autenticação e autorização
- API Management (APIM)
  - API gateway
  - Developer Portal (portal do desenvolvedor)
- Logic Apps (Workflow and Orchestration)
- Backend Systems
  - Azure services (ex.: Functions, Storage, Service Bus, Cosmos DB, SQL, etc.)
  - SaaS services (ex.: Dynamics 365, Salesforce, Office 365, ServiceNow, etc.)
  - Web services (REST e SOAP)
- Agrupamento e governança: Resource Group
- Controles de segurança e operação recomendados (não explícitos no diagrama, mas implicados)
  - Key Vault para segredos e certificados
  - Observabilidade: Azure Monitor, Log Analytics, Application Insights, Microsoft Sentinel, Defender for Cloud
  - Rede: VNet, Private Endpoints, NSG/Firewall, WAF (Front Door/App Gateway)

Interação entre os componentes:
- 0. Onboarding: Desenvolvedores usam o Developer Portal para descobrir APIs, ler documentação e registrar aplicações (criando client_id e obtendo segredos/chaves de subscrição, se habilitado).
- 1. Consumo: O cliente envia requisições HTTP para o API Gateway do APIM, incluindo token OAuth2/OIDC (Bearer) e/ou chave de subscrição.
- 2. Autenticação/Autorização: O APIM valida o token junto ao Microsoft Entra (issuer/audience/assinatura/exp) e aplica políticas de autorização (escopos/roles/claims). Opcionalmente mTLS.
- 3. Governança e políticas no APIM: Rate limiting/quotas, validação de schema (JSON/XML), transformação (REST ↔ SOAP), sanitização de headers, CORS, cache, roteamento.
- 4. Orquestração: O APIM encaminha a chamada para Logic Apps, que coordena fluxos, fan-out/fan-in, compensações, idempotência, e integra-se a conectores nativos.
- 5. Integração com Backends: Logic Apps (ou o próprio APIM, conforme a rota) consome:
  - Azure services via VNet e Private Endpoints
  - SaaS services via conectores/HTTP com OAuth

  - Web services REST/SOAP externos via HTTP(S)
  A resposta é agregada no Logic Apps (se aplicável) e retornada ao APIM, que a envia ao cliente.

O que esse sistema faz:
- Plataforma de exposição, governança e orquestração de APIs. O APIM publica interfaces seguras e documentadas para consumidores internos/externos; o Logic Apps compõe e automatiza processos entre serviços Azure, SaaS e web services (REST/SOAP). É um padrão de integração/ESB leve com API Gateway na borda.

Vulnerabilidades e Solução para cada vulnerabilidade:
- Token forjado ou inválido (Spoofing)
  - Solução: Validar OIDC no APIM (issuer, audience, assinatura via JWKS, exp/nbf); exigir OAuth2 com escopos/roles; considerar mTLS para parceiros críticos; certificate pinning no cliente.
- Quebra de autorização (Acesso além do permitido)
  - Solução: Autorização baseada em claims/roles no APIM e nos backends; segmentação por produto no APIM; RBAC no Entra; separar chaves por consumidor; princípio de privilégio mínimo.
- Injeções (SQL/NoSQL/Command/XML)
  - Solução: Validação de entrada no APIM (JSON/XML schema), encoding adequado; parametrização em queries; desabilitar concatenação dinâmica; WAF na borda.
- XXE e ataques SOAP/XML
  - Solução: Desabilitar DTD/entidades externas; usar transformações seguras (xml-to-json) no APIM; validar schemas e tamanhos; limitar profundidade e tamanho de XML.
- Manipulação de payload/headers em trânsito (Tampering)
  - Solução: TLS 1.2+ em todo o caminho; HSTS; remoção de headers sensíveis; assinatura de mensagens para integrações críticas; integrity checks.
- Replay de requisições
  - Solução: Exigir exp/iat nos tokens; idempotency keys no APIM/Logic Apps; nonce/timestamps; janelas de tempo curtas; rate limit por chave/identidade.
- DoS/DDoS e abuso de APIs
  - Solução: Azure DDoS Protection + WAF; rate limiting e quotas no APIM; circuit breaker e timeouts; cache de respostas; autoscaling onde aplicável.
- Armazenamento de segredos em texto claro (APIM/Logic Apps)
  - Solução: Managed Identity e Key Vault para segredos/certificados; desabilitar segredos no código; rotação periódica; purge protection e soft delete no Key Vault.
- Exposição indevida via CORS
  - Solução: Restringir origins/domínios; métodos/headers mínimos; sem wildcard em produção; validar credenciais CORS.
- Vazamento de dados em logs/traces
  - Solução: Mascarar PII/segredos no APIM e Logic Apps; controles de retenção; acesso a logs via RBAC/Privileged Identity Management; classificação de dados.
- Falta de isolamento de rede
  - Solução: APIM em modo VNet/Private; Logic Apps Standard com VNet Integration; Private Endpoints para backends; NSG/Azure Firewall; desabilitar acesso público quando possível.
- Portal do desenvolvedor exposto/abusado
  - Solução: SSO com Entra; aprovação manual de subscrições; reCAPTCHA/anti-automation; limitação de auto-registro; WAF e monitoramento.
- Erros e exceções verborrágicas
  - Solução: Customizar mensagens de erro no APIM; não vazar stack traces; correlação via IDs.
- Dependências/TLS fracos
  - Solução: Impor TLS 1.2/1.3; desabilitar ciphers fracos; política de atualização de conectores/serviços; verificações de baseline do Defender for Cloud.
- Duplicidade em fluxos assíncronos
  - Solução: Idempotência no Logic Apps (chave de negócio), filas com deduplicação, transações compensatórias.

Gere um Relatório de Modelagem de Ameaças, baseado na metodologia STRIDE:
- Escopo
  - Exposição e consumo de APIs públicas/privadas via Azure API Management, com orquestração por Logic Apps e integrações com Azure services, SaaS e web services (REST/SOAP). Inclui Developer Portal e Microsoft Entra ID.
- Ativos principais
  - APIs e contratos (OpenAPI), tokens e segredos, dados de negócio, logs/telemetria, pipelines/artefatos de deploy, identidades (usuários/app registrations), certificados.
- Fronteiras de confiança
  - Internet ↔ APIM (borda)
  - APIM ↔ Microsoft Entra (controle de identidade)
  - APIM ↔ Logic Apps (orquestração)
  - Logic Apps/APIM ↔ Backends (Azure/SaaS/Web)
  - Usuários ↔ Developer Portal

  - Plano de gerenciamento (Azure Portal/ARM) ↔ Recursos
- Fluxos de dados (referência aos números do diagrama)
  - DF1: Cliente → APIM (HTTP)
  - DF2: APIM ↔ Entra (validação de token)
  - DF3: APIM aplica políticas/roteia
  - DF4: APIM → Logic Apps (orquestração)
  - DF5: Logic Apps/APIM → Backends (Azure/SaaS/REST/SOAP)
- STRIDE por categoria (exemplos e controles)
  - Spoofing
    - Riscos: Tokens falsos, mTLS ausente, app registrations comprometidas.
    - Controles: OIDC estrito no APIM; mTLS para parceiros; Managed Identity; Conditional Access; PIM para admins.
  - Tampering
    - Riscos: Payload/headers alterados; manipulação em transformações (REST↔SOAP).
    - Controles: TLS forte; schema validation; políticas de tamanho/limite; assinatura de mensagens quando aplicável.
  - Repudiation
    - Riscos: Consumidores negam chamadas; integrações sem trilha.
    - Controles: Logging imutável (Log Analytics/Sentinel), correlação (x-correlation-id), retenção e carimbo de tempo, exportação para armazenamento WORM quando necessário.
  - Information Disclosure
    - Riscos: PII em logs; erros verborrágicos; CORS amplo; conexões públicas.
    - Controles: Mascaramento em logs; mensagens de erro customizadas; CORS restrito; Private Endpoints/VNet; DLP para conectores.
  - Denial of Service
    - Riscos: Flood de requisições; picos em conectores SaaS; dependências lentas.
    - Controles: DDoS Protection + WAF; rate limit/quotas/cota por produto; cache; circuit breaker e timeout; filas/buffers; autoscaling.
  - Elevation of Privilege
    - Riscos: Políticas APIM permissivas; roles mal configurados; credenciais privilegiadas expostas.
    - Controles: RBAC mínimo; validação de claims/escopos por rota; segregar produtos/ambientes; Key Vault; revisões de acesso periódicas; políticas de branch e CI/CD com aprovações.
- Riscos priorizados (exemplo)
  - Alto: Falta de validação OIDC no APIM; ausência de rate limiting; segredos fora do Key Vault; backends sem Private Endpoint.
  - Médio: CORS amplo; logs com PII; erros verborrágicos; falta de idempotência.
  - Baixo: Ciphers herdados habilitados; headers informativos sobrando.
- Plano de mitigação resumido
  - 0–30 dias: Habilitar validação OIDC e políticas essenciais no APIM (rate limit, schema validation, CORS); mover segredos para Key Vault; ligar diagnostic logs; endurecer TLS.
  - 30–90 dias: VNet/Private Endpoints; WAF + DDoS; Managed Identity em Logic Apps; mascaramento de logs; dashboards e alertas; idempotência.
  - 90–180 dias: Hardening do Developer Portal; revisão de papéis/entitlements; testes de carga e chaos; runbooks de resposta a incidentes; revisão de conectores e DLP.

Additional resourcing needs:
- Papéis a envolver
  - Project Manager do STRIDE (coordenação e escopo)
  - Network Administrator (VNet, Private Endpoints, NSG, Firewall, WAF, DDoS)
  - Security Officer/CISO (padrões de segurança, políticas, exceções)
  - Cloud/DevOps (arquitetura Azure, IaC, CI/CD, observabilidade)
  - Software Developer/API Owner (contratos OpenAPI, versionamento, políticas do APIM)
  - Integration Engineer/Logic Apps (orquestração, conectores, idempotência)
  - Data Protection Officer (privacidade/PII, retenção/masking)
  - Compliance/Governance (auditoria, trilhas, SoD)
  - Opcional: Hardware/IoT Developer (se houver dispositivos/segredos embarcados)
- Termos de referência (para o workshop STRIDE)
  - Definir escopo e fronteiras do sistema
  - Identificar stakeholders e cadeia de fornecedores
  - Levantar diagramas/artefatos (UML/DFD, inventário de APIs, fluxos, políticas APIM)
  - Identificar ameaças por STRIDE, avaliar probabilidade/impacto e plano de tratamento
  - Manutenção contínua: revisar a cada mudança relevante e periodicamente

Observações finais:
- Este desenho representa um padrão de integração com API Gateway (APIM) + orquestração (Logic Apps) em Azure, expondo APIs para consumo seguro e compondo serviços Azure, SaaS e web (REST/SOAP). O fortalecimento de identidade (Entra), rede (VNet/Private Endpoints/WAF), segredos (Key Vault) e governança de APIs (políticas APIM) são os pilares de segurança recomendados.

Sugestões de melhorias

A seguir estão:
1) o diagrama mermaid com componentes, relações, fronteiras e anotações de controles corretivos;
2) um script Terraform inicializável (IaC) para criar a fundação desta arquitetura no Azure (com APIM, Logic Apps Standard, VNet com DDoS, Key Vault, Observabilidade, políticas essenciais do APIM e exemplos de conexão a backends via Private Endpoint);
3) o Relatório de Modelagem de Ameaças (STRIDE) consolidado.

Diagrama (Mermaid)

```mermaid
flowchart TB
  %% Fronteiras
  subgraph Internet[Internet / Parceiros / Apps Móveis e Web]
    C[Consumidor/Cliente de API]
    DP[Developer Portal (APIM)]
  end

  subgraph Entra[Microsoft Entra ID (Azure AD)]
    OIDC[(OIDC/OAuth2: Authority, JWKS, Scopes, Roles)]
  end

  subgraph AzureSub[Azure Subscription - Resource Group]
    direction TB

    subgraph Gov[Observabilidade e Segurança]
      MON[Azure Monitor / Log Analytics / App Insights]
      SENT[Microsoft Sentinel]
      DEF[Defender for Cloud]
      KV[(Key Vault: segredos/certs)]
    end

    subgraph Net[Virtual Network (VNet) + DDoS]
      DDoS[DDoS Protection]
      subgraph SegPriv[Zonas Privadas / PE]
        PDNS[Private DNS Zones]
      end

      subgraph AppTier[Zona de Aplicação]
        LA[Logic Apps Standard (orquestração)]
        APIM[(API Management - API Gateway)]
      end

      subgraph Edge[Opicional: WAF na borda]
        WAF[App Gateway/Front Door (WAF)]
      end
    end

    subgraph Backends[Backends e Integrações]
      direction TB
      AZS[Azure Services (Functions, Storage, Service Bus, Cosmos DB, SQL...)]
      SAAS[SaaS (Dynamics 365, Salesforce, O365, ServiceNow...)]
      WS[Web Services REST/SOAP externos]
    end
  end

  %% Fluxos de Dados
  C -- "DF1: HTTPS + Bearer (OIDC) + Sub key (opcional) + mTLS (opcional)" -->|Consumo| WAF
  WAF -->|Roteia/Protege| APIM
  C --- DP
  APIM <-- "DF2: Validação de token (issuer/aud/assinatura)" --> OIDC
  APIM -->|DF3: Políticas (rate limit, schema, CORS, cache, transformação)| APIM
  APIM -->|DF4: Roteia/orquestra| LA
  LA -->|DF5: PE/VNet/OAuth| AZS
  LA -->|DF5: Conectores SaaS / OAuth| SAAS
  LA -->|DF5: HTTP(S) seguro / mTLS| WS
  APIM -->|Logs/Telemetria| MON
  LA -->|Logs/Telemetria| MON
```

```
  MON --> SENT
  AzureSub -.Segredos/Certs.-> KV
  Net --- DDoS
  SegPriv --- PDNS

  %% Notas de Controles (correções de vulnerabilidades)
  classDef ctrl fill:#e3f7ea,stroke:#0f7b45,color:#0f7b45;
  R1[Controles: TLS1.2+/HSTS, WAF + DDoS, Rate limiting/Quotas]:::ctrl
  R2[Controles: OIDC estrito (issuer/aud/exp/nbf), mTLS opcional]:::ctrl
  R3[Controles: Schema validation, CORS restrito, sanitização de headers]:::ctrl
  R4[Controles: Idempotência, correlação, retries/timeout/circuit breaker]:::ctrl
  R5[Controles: Managed Identity + Key Vault, PE/VNet/NSG/Firewall]:::ctrl
  R6[Controles: Mascaramento de logs, erros customizados, RBAC/PIM]:::ctrl

  C -.-> R1
  APIM -.-> R2
  APIM -.-> R3
  LA -.-> R4
  Backends -.-> R5
  Gov -.-> R6
```

Principais correções aplicadas e onde
- Identidade e autorização: validação OIDC completa no APIM, escopos/roles por rota; opção de mTLS para parceiros sensíveis.
- Governaça de APIs: rate limiting e quotas; validação de schema JSON/XML; transformação e sanitização de headers; CORS restrito; cache quando aplicável.
- Rede: VNet, DDoS Standard, WAF na borda (Front Door ou App Gateway), Private Endpoints e Private DNS para serviços PaaS; NSG/Firewall onde cabível.
- Segredos: Managed Identity e Key Vault com purge protection/soft-delete e rotação; sem segredos em código.
- Observabilidade segura: Logs para Log Analytics/App Insights; mascaramento/PII, mensagens de erro não verborrágicas; Sentinel e Defender for Cloud.
- Resiliência: timeouts, circuit breaker, idempotência no Logic Apps; filas/buffers quando necessário.

Terraform (Azure)
Observações
- É um template inicial com boas práticas. Ajuste nomes, SKUs, domínios e integrações.
- APIM em modo External VNet (mais simples). Para total isolamento, troque para Internal e posicione um WAF (App Gateway/Front Door) na borda.
- Inclui: RG, VNet+Subnets, DDoS, APIM (Premium) com políticas globais (OIDC, rate limit, CORS, headers), Logic Apps Standard em VNet, Key Vault, Log Analytics + App Insights, Sentinel/Defender, Storage com Private Endpoint e Private DNS.
- Alguns valores sensíveis devem ir para um cofre de segredos/Key Vault/variáveis protegidas no pipeline.

Código
```hcl
terraform {
  required_version = ">= 1.5.0"
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = ">= 3.108.0"
    }
    azuread = {
      source  = "hashicorp/azuread"
      version = ">= 2.50.0"
    }
  }
}

provider "azurerm" {
  features {}
}

# ============================
# Variáveis
# ============================
```

```
variable "prefix" {
  description = "Prefixo de nomes para recursos"
  type        = string
  default     = "apiesb"
}

variable "location" {
  description = "Região Azure"
  type        = string
  default     = "eastus"
}

variable "tenant_id" {
  description = "Tenant ID do Microsoft Entra"
  type        = string
}

variable "apim_sku_name" {
  description = "SKU do APIM (Premium recomendado para VNet)"
  type        = string
  default     = "Premium_1"
}

variable "allowed_cors_origins" {
  description = "Lista de origens permitidas no CORS (sem wildcard em produção)"
  type        = list(string)
  default     = ["https://app.exemplo.com"]
}

variable "logicapp_sku" {
  description = "SKU do Logic App Standard"
  type        = string
  default     = "WS1"
}

# Opcional: URL do endpoint do Logic App para exemplo de backend no APIM
variable "logicapp_callback_url" {
  description = "Callback URL de um workflow HTTP-trigger do Logic App (opcional para demonstrar
backend no APIM)"
  type        = string
  default     = ""
}

# ============================
# Resource Group
# ============================
resource "azurerm_resource_group" "rg" {
  name     = "${var.prefix}-rg"
  location = var.location
}

# ============================
# Observabilidade
# ============================
resource "azurerm_log_analytics_workspace" "law" {
  name                = "${var.prefix}-law"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  sku                 = "PerGB2018"
  retention_in_days   = 30
}

resource "azurerm_application_insights" "appi" {
  name                = "${var.prefix}-appi"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  application_type    = "web"
  workspace_id        = azurerm_log_analytics_workspace.law.id
}
```

```
# Sentinel
resource "azurerm_sentinel_log_analytics_workspace_onboarding" "sentinel" {
  workspace_id = azurerm_log_analytics_workspace.law.id
}

# Defender for Cloud (básico para API/Compute/Storage)
resource "azurerm_security_center_subscription_pricing" "defender_api" {
  tier          = "Standard"
  resource_type = "AppServices"
}
resource "azurerm_security_center_subscription_pricing" "defender_storage" {
  tier          = "Standard"
  resource_type = "StorageAccounts"
}
resource "azurerm_security_center_subscription_pricing" "defender_sql" {
  tier          = "Standard"
  resource_type = "SqlServers"
}

# ============================
# Rede: VNet + DDoS + Subnets + Private DNS
# ============================
resource "azurerm_network_ddos_protection_plan" "ddos" {
  name                = "${var.prefix}-ddos"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
}

resource "azurerm_virtual_network" "vnet" {
  name                = "${var.prefix}-vnet"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  address_space       = ["10.10.0.0/16"]

  ddos_protection_plan {
    id     = azurerm_network_ddos_protection_plan.ddos.id
    enable = true
  }
}

resource "azurerm_subnet" "subnet_apim" {
  name                 = "snet-apim"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.1.0/24"]
}

resource "azurerm_subnet" "subnet_logicapps" {
  name                 = "snet-logicapps"
  resource_group_name  = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.2.0/24"]
  delegation {
    name = "delegation"
    service_delegation {
      name = "Microsoft.Web/serverFarms"
      actions = [
        "Microsoft.Network/virtualNetworks/subnets/action",
      ]
    }
  }
}

resource "azurerm_subnet" "subnet_pe" {
  name                                      = "snet-private-endpoints"
  resource_group_name                       = azurerm_resource_group.rg.name
  virtual_network_name                      = azurerm_virtual_network.vnet.name
  address_prefixes                          = ["10.10.3.0/24"]
```

```
  private_endpoint_network_policies_enabled = false
}

# Private DNS Zone para Storage (exemplo)
resource "azurerm_private_dns_zone" "pdns_blob" {
  name                = "privatelink.blob.core.windows.net"
  resource_group_name = azurerm_resource_group.rg.name
}

resource "azurerm_private_dns_zone_virtual_network_link" "pdns_blob_vnet" {
  name                  = "${var.prefix}-pdns-blob-link"
  resource_group_name   = azurerm_resource_group.rg.name
  private_dns_zone_name = azurerm_private_dns_zone.pdns_blob.name
  virtual_network_id    = azurerm_virtual_network.vnet.id
}

# ============================
# Key Vault (segredos/certs)
# ============================
resource "azurerm_key_vault" "kv" {
  name                        = "${var.prefix}kv${substr(replace(uuid(), "-", ""), 0, 6)}"
  location                    = azurerm_resource_group.rg.location
  resource_group_name         = azurerm_resource_group.rg.name
  tenant_id                   = var.tenant_id
  sku_name                    = "standard"
  purge_protection_enabled    = true
  soft_delete_retention_days  = 90
  enable_rbac_authorization   = true
}

# ============================
# APIM
# ============================
resource "azurerm_api_management" "apim" {
  name                = "${var.prefix}-apim"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  publisher_name      = "API Team"
  publisher_email     = "api-team@example.com"
  sku_name            = var.apim_sku_name

  identity {
    type = "SystemAssigned"
  }

  # VNet: External (mantém endpoint público + injeção de VNet)
  virtual_network_type = "External"
  virtual_network_configuration {
    subnet_id = azurerm_subnet.subnet_apim.id
  }

  protocols {
    enable_http2 = true
  }

  # Hardening de TLS
  custom_properties = {
    "Microsoft.WindowsAzure.ApiManagement.Gateway.Security.Protocols.Tls10"       = "False"
    "Microsoft.WindowsAzure.ApiManagement.Gateway.Security.Protocols.Tls11"       = "False"
    "Microsoft.WindowsAzure.ApiManagement.Gateway.Security.Protocols.Ssl30"       = "False"
    "Microsoft.WindowsAzure.ApiManagement.Gateway.Security.Backend.Protocols.Tls10" = "False"
    "Microsoft.WindowsAzure.ApiManagement.Gateway.Security.Backend.Protocols.Tls11" = "False"
  }
}

# Provider OIDC do Microsoft Entra para validação de JWT no APIM
resource "azurerm_api_management_openid_connect_provider" "oidc" {
  name                = "entra-oidc"
  api_management_name = azurerm_api_management.apim.name
```

```
  resource_group_name = azurerm_resource_group.rg.name

  display_name      = "Microsoft Entra ID"
  metadata_endpoint =
"https://login.microsoftonline.com/${var.tenant_id}/v2.0/.well-known/openid-configuration"
  client_id         = "REPLACE_WITH_APP_CLIENT_ID" # Ajuste: App Registration do cliente do
portal se necessário
  # client_secret   = "Use Named Value + Key Vault; evite segredos no estado"
}

# Named Value para audiência esperada (ex.: API App ID URI)
resource "azurerm_api_management_named_value" "audience" {
  name                = "jwt-expected-audience"
  api_management_name = azurerm_api_management.apim.name
  resource_group_name = azurerm_resource_group.rg.name
  display_name        = "JWT Expected Audience"
  value               = "api://example-audience" # Ajuste para seu App ID URI / audience
}

# Política global do APIM (segurança: OIDC, rate limit, CORS, sanitização, erros)
resource "azurerm_api_management_policy" "apim_policy" {
  api_management_id = azurerm_api_management.apim.id
  xml_content = <<POL
<policies>
  <inbound>
    <base />
    <!-- CORS estrito -->
    <cors allow-credentials="false">
      <allowed-origins>
        % for origin in terraform.get("allowed_cors_origins", []) %
        <origin>${origin}</origin>
        % endfor %
      </allowed-origins>
      <allowed-methods preflight-result-max-age="300">
        <method>GET</method>
        <method>POST</method>
        <method>PUT</method>
        <method>DELETE</method>
      </allowed-methods>
      <allowed-headers>
        <header>accept</header>
        <header>content-type</header>
        <header>authorization</header>
      </allowed-headers>
      <expose-headers>
        <header>x-correlation-id</header>
      </expose-headers>
    </cors>

    <!-- Correlation ID -->
    <set-header name="x-correlation-id" exists-action="override">
      <value>@(context.RequestId)</value>
    </set-header>

    <!-- Sanitização de headers sensíveis -->
    <set-header name="Server" exists-action="delete" />
    <set-header name="X-Powered-By" exists-action="delete" />

    <!-- Validação OIDC -->
    <validate-jwt header-name="Authorization" failed-validation-httpcode="401"
require-expiration-time="true" require-signed-tokens="true" clock-skew="60">
      <openid-config
url="@{${azurerm_api_management_openid_connect_provider.oidc.metadata_endpoint}}" />
      <audiences>
        <audience>@{${azurerm_api_management_named_value.audience.value}}</audience>
      </audiences>
    </validate-jwt>

    <!-- Rate limiting / quota básica (ajustar por produto) -->
```

```
    <rate-limit calls="100" renewal-period="60" />
  </inbound>
  <backend>
    <base />
  </backend>
  <outbound>
    <base />
    <!-- Mascaramento/remoção de informações -->
    <set-header name="X-AspNet-Version" exists-action="delete" />
    <set-header name="X-Powered-By" exists-action="delete" />
  </outbound>
  <on-error>
    <base />
    <!-- Erros não verborrágicos -->
    <return-response>
      <set-status code="500" reason="Internal Server Error" />
      <set-header name="Content-Type" exists-action="override">
        <value>application/json</value>
      </set-header>
      <set-body>{"error":"An unexpected error
occurred.","correlationId":"@(context.RequestId)"}</set-body>
    </return-response>
  </on-error>
</policies>
POL
}

# Exemplo (opcional): Backend no APIM apontando para um Logic App HTTP-trigger
resource "azurerm_api_management_backend" "logicapp_backend" {
  count               = length(var.logicapp_callback_url) > 0 ? 1 : 0
  name                = "logicapp-backend"
  api_management_name = azurerm_api_management.apim.name
  resource_group_name = azurerm_resource_group.rg.name
  protocol            = "http"
  url                 = var.logicapp_callback_url
}

resource "azurerm_api_management_api" "sample_api" {
  name                = "sample-orchestration"
  resource_group_name = azurerm_resource_group.rg.name
  api_management_name = azurerm_api_management.apim.name
  revision            = "1"
  display_name        = "Sample Orchestration API"
  path                = "orchestration"
  protocols           = ["https"]
}

resource "azurerm_api_management_api_operation" "op_ping" {
  operation_id        = "ping"
  api_name            = azurerm_api_management_api.sample_api.name
  api_management_name = azurerm_api_management.apim.name
  resource_group_name = azurerm_resource_group.rg.name
  display_name        = "Ping"
  method              = "GET"
  url_template        = "/ping"
  response {
    status = 200
    description = "ok"
  }
}

# ============================
# Logic Apps Standard (em VNet)
# ============================
resource "azurerm_storage_account" "las" {
  name                     = "${var.prefix}las${substr(replace(uuid(), "-", ""), 0, 6)}"
  resource_group_name      = azurerm_resource_group.rg.name
  location                 = azurerm_resource_group.rg.location
  account_tier             = "Standard"
```

```
  account_replication_type = "LRS"
  min_tls_version          = "TLS1_2"
  allow_nested_items_to_be_public = false
}

resource "azurerm_service_plan" "asp" {
  name                = "${var.prefix}-asp"
  resource_group_name = azurerm_resource_group.rg.name
  location            = azurerm_resource_group.rg.location
  os_type             = "Windows"
  sku_name            = var.logicapp_sku
  worker_count        = 1
}

resource "azurerm_logic_app_standard" "la" {
  name                       = "${var.prefix}-la"
  location                   = azurerm_resource_group.rg.location
  resource_group_name        = azurerm_resource_group.rg.name
  app_service_plan_id        = azurerm_service_plan.asp.id
  storage_account_name       = azurerm_storage_account.las.name
  storage_account_access_key = azurerm_storage_account.las.primary_access_key
  virtual_network_subnet_id  = azurerm_subnet.subnet_logicapps.id
  identity {
    type = "SystemAssigned"
  }
  app_settings = {
    "APPINSIGHTS_INSTRUMENTATIONKEY" =
azurerm_application_insights.appi.instrumentation_key
    "APPLICATIONINSIGHTS_CONNECTION_STRING" =
azurerm_application_insights.appi.connection_string
    "WEBSITE_RUN_FROM_PACKAGE" = "0"
  }
}

# ============================
# Private Endpoint de exemplo (Storage)
# ============================
resource "azurerm_private_endpoint" "pe_blob" {
  name                = "${var.prefix}-pe-blob"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  subnet_id           = azurerm_subnet.subnet_pe.id

  private_service_connection {
    name                           = "${var.prefix}-pe-blob-conn"
    private_connection_resource_id = azurerm_storage_account.las.id
    is_manual_connection           = false
    subresource_names              = ["blob"]
  }
}

resource "azurerm_private_dns_a_record" "pe_blob_dns" {
  name                = azurerm_storage_account.las.name
  zone_name           = azurerm_private_dns_zone.pdns_blob.name
  resource_group_name = azurerm_resource_group.rg.name
  ttl                 = 300
  records             =
[azurerm_private_endpoint.pe_blob.private_service_connection[0].private_ip_address]
}

# ============================
# Diagnostic Settings
# ============================
resource "azurerm_monitor_diagnostic_setting" "diag_apim" {
  name                       = "${var.prefix}-diag-apim"
  target_resource_id         = azurerm_api_management.apim.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  enabled_log {
```

```
    category = "GatewayLogs"
  }
  enabled_log {
    category = "WebSocketConnectionLogs"
  }
  enabled_log {
    category = "AuditLogs"
  }
  metric {
    category = "AllMetrics"
  }
}

resource "azurerm_monitor_diagnostic_setting" "diag_la" {
  name                       = "${var.prefix}-diag-la"
  target_resource_id         = azurerm_logic_app_standard.la.id
  log_analytics_workspace_id = azurerm_log_analytics_workspace.law.id

  metric {
    category = "AllMetrics"
  }
}

# ============================
# Saídas
# ============================
output "apim_gateway_url" {
  value = azurerm_api_management.apim.gateway_regional_url
}

output "logic_app_default_hostname" {
  value = azurerm_logic_app_standard.la.default_hostname
}

output "key_vault_name" {
  value = azurerm_key_vault.kv.name
}
```

Próximos passos (infra e políticas)
- Se desejar APIM totalmente privado: mude azurerm_api_management.virtual_network_type para Internal e posicione um WAF (App Gateway v2 ou Front Door Premium) na borda. Configure listeners, certificados e roteamento para o APIM interno.
- Configure Produtos e Subscrições no APIM com quotas/rate por consumidor. Separe por ambiente (dev/test/prod) e por audiência (interno/externo).
- Políticas adicionais por API/operação: validação de schema (validate-content), transformação xml-to-json/json-to-xml, cache-lookup/store, chaves de idempotência.
- mTLS: carregue certificados de cliente no APIM (Key Vault) e exija <check-header name="X-ARR-ClientCert"> ou política <authenticate-client-certificate>.
- Backends: crie Private Endpoints para Cosmos, SQL, Service Bus etc., com Private DNS Zones correspondentes. Use Managed Identity do Logic Apps/APIM para autenticação de backends Azure.
- Developer Portal: habilite SSO com Entra; limite auto-registro; modere subscrições; anti-automation (Captcha) por WAF.

Relatório de Modelagem de Ameaças (STRIDE)

Escopo
- Exposição e consumo de APIs públicas/privadas via Azure API Management, com orquestração por Logic Apps e integrações com Azure services, SaaS e web services (REST/SOAP). Inclui Developer Portal e Microsoft Entra ID.

Ativos principais
- APIs/contratos (OpenAPI), tokens e segredos, dados de negócio, logs/telemetria, pipelines/artefatos de deploy, identidades (usuários/app registrations), certificados.

Fronteiras de confiança
- Internet ↔ APIM (borda)
- APIM ↔ Microsoft Entra (controle de identidade)
- APIM ↔ Logic Apps (orquestração)

- Logic Apps/APIM ↔ Backends (Azure/SaaS/Web)
- Usuários ↔ Developer Portal
- Plano de gerenciamento (Azure Portal/ARM) ↔ Recursos

Fluxos de dados
- DF1: Cliente → APIM (HTTPS, Bearer, chave de subscrição, mTLS opcional)
- DF2: APIM ↔ Entra (validação de token OIDC)
- DF3: APIM aplica políticas e roteia
- DF4: APIM → Logic Apps (orquestração)
- DF5: Logic Apps/APIM → Backends (Azure/SaaS/REST/SOAP, via VNet/PE e/ou OAuth)

STRIDE por categoria (ameaças e controles)
- Spoofing
  - Riscos: Tokens falsos, mTLS ausente, app registrations comprometidas.
  - Controles: Validação OIDC estrita no APIM (issuer/audience/assinatura/exp/nbf); exigir OAuth2 com escopos/roles; mTLS para parceiros; Managed Identity; Conditional Access; PIM para admins.
- Tampering
  - Riscos: Payload/headers manipulados; transformação insegura REST↔SOAP/XML.
  - Controles: TLS 1.2+ e HSTS; schema validation e limites de tamanho/profundidade; sanitização de headers; assinatura de mensagens quando aplicável.
- Repudiation
  - Riscos: Consumidores negam chamadas; integrações sem trilha.
  - Controles: Logging central (Log Analytics/Sentinel), correlação (x-correlation-id), retenção e carimbo de tempo; exportação WORM quando necessário.
- Information Disclosure
  - Riscos: PII em logs; mensagens de erro verborrágicas; CORS amplo; endpoints públicos inadvertidos.
  - Controles: Mascaramento de logs; erros customizados no APIM; CORS restrito; Private Endpoints/VNet; DLP nos conectores; RBAC em dados.
- Denial of Service
  - Riscos: Flood/abuso de APIs; picos em conectores; dependências lentas.
  - Controles: DDoS Protection + WAF; rate limiting/quotas por produto/consumidor; cache; timeouts e circuit breaker; filas/buffers; autoscaling.
- Elevation of Privilege
  - Riscos: Políticas APIM permissivas; roles mal configurados; credenciais privilegiadas expostas.
  - Controles: RBAC de mínimo privilégio; validação de claims/escopos por rota; segregação por produtos/ambientes; Key Vault; revisões periódicas de acesso; políticas de branch e aprovações em CI/CD.

Riscos priorizados
- Alto: Falta de validação OIDC no APIM; ausência de rate limiting/quotas; segredos fora do Key Vault; backends PaaS sem Private Endpoint.
- Médio: CORS amplo; logs com PII; erros verborrágicos; ausência de idempotência.
- Baixo: Ciphers herdados habilitados; headers informativos sobrando.

Plano de mitigação
- 0–30 dias: Habilitar validação OIDC e políticas essenciais no APIM (rate limit, schema validation, CORS); mover segredos para Key Vault; ligar diagnostic logs; endurecer TLS.
- 30–90 dias: VNet/Private Endpoints; WAF + DDoS; Managed Identity no Logic Apps/APIM; mascaramento de logs; dashboards e alertas; idempotência.
- 90–180 dias: Hardening do Developer Portal; revisão de papéis/entitlements; testes de carga/chaos; runbooks de IR; revisão de conectores e DLP.

Recursos e papéis
- Project Manager STRIDE, Network Administrator, Security Officer/CISO, Cloud/DevOps, Software Developer/API Owner, Integration Engineer/Logic Apps, Data Protection Officer, Compliance/Governance, opcional Hardware/IoT.

Termos de referência (workshop STRIDE)
- Definir escopo e fronteiras; stakeholders e cadeia de fornecedores; diagramas/artefatos (UML/DFD, inventário de APIs, fluxos, políticas APIM); ameaças por STRIDE com prob/impacto; plano de tratamento e revisão contínua.

Observações finais
- Este padrão APIM + Logic Apps em Azure fortalece identidade (Entra), rede (VNet/PE/WAF/DDoS), segredos (Key Vault) e governança (políticas APIM). Ajuste o nível de isolamento (APIM Interno + WAF) conforme o risco e a exposição necessária.