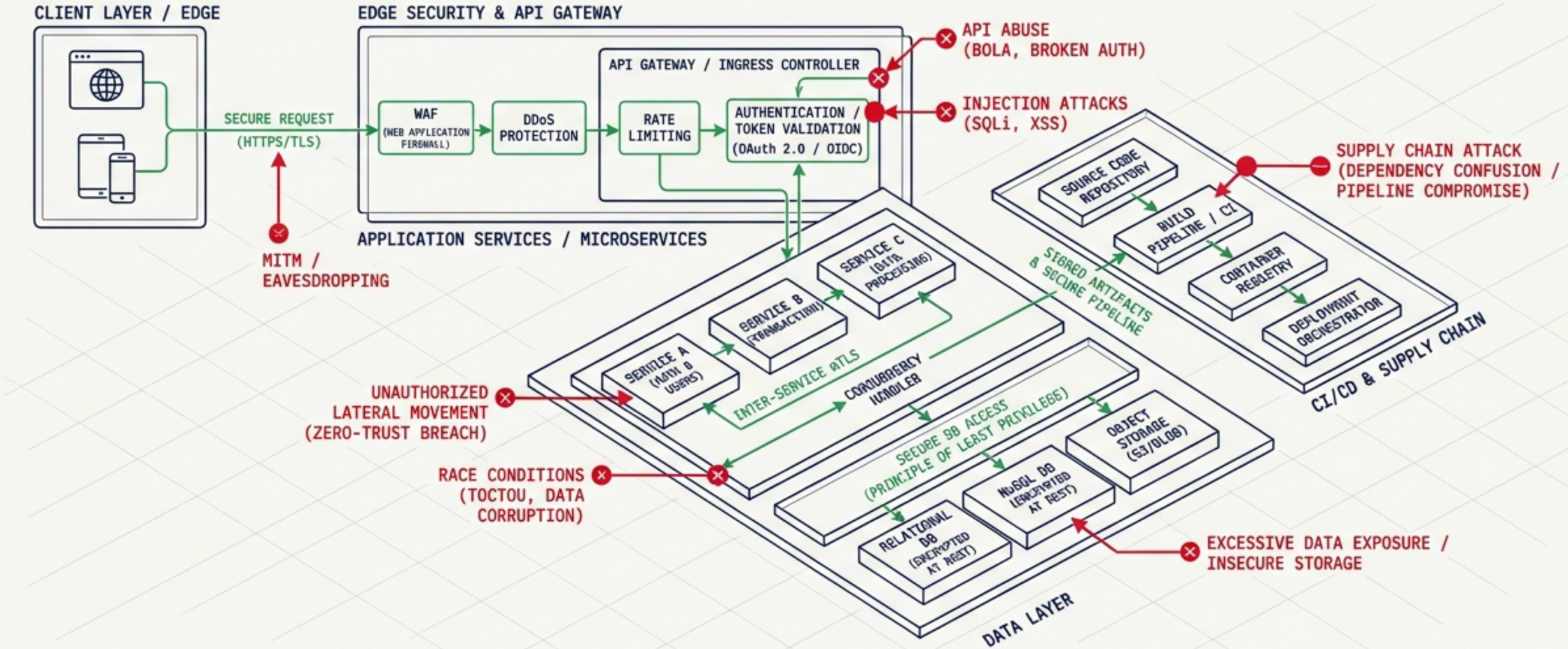
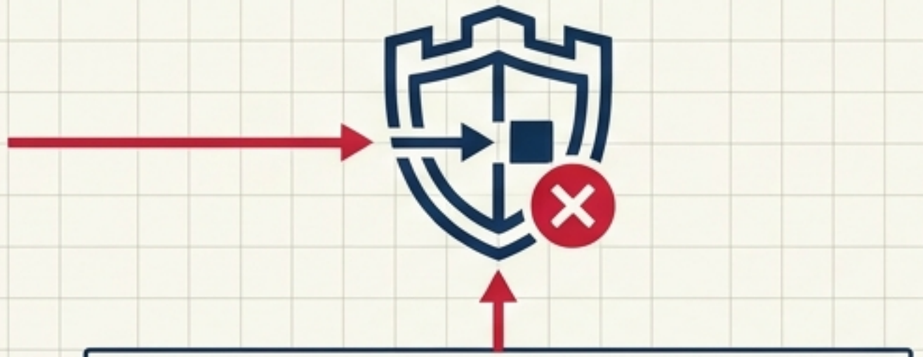
WEB SECURITY — S03E06: API AUTHORIZATION, CONCURRENCY, SUPPLY CHAIN & ZERO-TRUST DEFENSE-IN-DEPTH

The Capstone Architecture Blueprint for Senior Engineers



INTEGRATING OWASP API TOP 10, NIST SP 800-207, AND RFC 9116

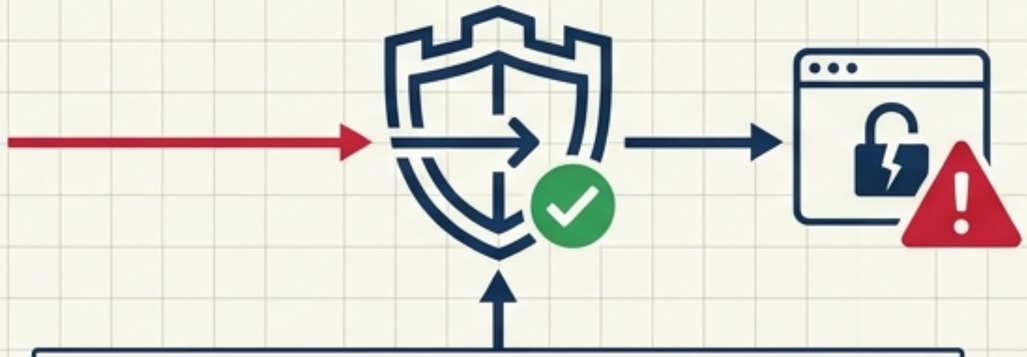
Gemini Notebook



```
{
  {"acct_id": "' OR 1=1--"}
}
```

SYNTAX ERROR CAUGHT

WAF detects and blocks known attack patterns in request syntax.



```
{
  {"acct_id": 901}
}
```

(altered from user's actual ID of 523)

SEMANTIC BLINDSPOT: LOGIC ERROR MISSED

WAF permits syntactically valid requests that abuse application logic.

Authentication validates identity
(Who are you?)

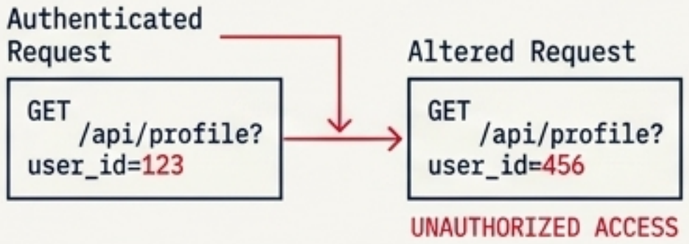
Authorization validates context
(Are you allowed to access resource #901?)

Perimeter defenses validate structure. Syntactically clean requests routinely abuse missing state transitions, rendering scanners blind to business-logic breaches.

Gemini Notebook

API1: Broken Object Level Auth (BOLA)

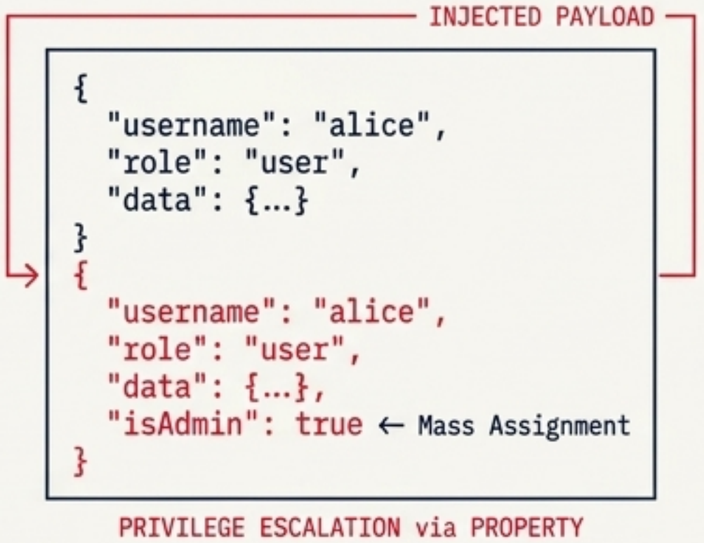
Horizontal Privilege Escalation (IDOR)



An authenticated user alters a lookup ID to access peer resources.

API3: Broken Object Property Level Auth (BOPLA)

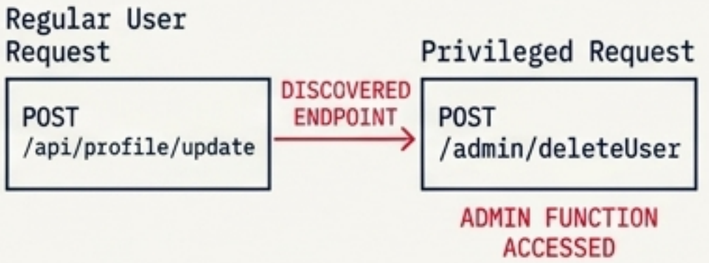
Mass Assignment / Excessive Data Exposure



User injects restricted properties which the framework implicitly binds to the database model.

API5: Broken Function Level Auth (BFLA)

Vertical Privilege Escalation



A non-administrative user discovers and successfully invokes a privileged endpoint.

Gemini Notebook

COMPARISON MATRIX: ACCESS CONTROL MODELS (RBAC vs. ABAC vs. ReBAC)

Dimension	RBAC (Role-Based)	ABAC (Attribute-Based)	ReBAC (Relationship-Based)
Granularity	Coarse (Roles only)	Fine (Time, IP, Device, Status)	Ultra-fine (Relationships between specific resources)
Speed / Role Explosion	High risk of role explosion causing latency	Fast, but policy evaluation can be complex	Graph-based, optimized for complex hierarchical queries
Multi-Tenancy	Poor (requires duplicated rules per tenant)	Excellent (Context-aware)	Excellent (Native scoping)

Prefer ABAC and ReBAC. In RBAC, access decisions are made on the mere presence of roles, failing to support object-level constraints or multi-factor logic required for modern SaaS architectures. Utilize centralized policy engines (OPA/Cedar) and Zanzibar-style ReBAC graphs.

Gemini Notebook

Server issues a UUID
123e4567-e89b-42d3-a456-556642440000

Attacker captures UUID from public forum, review, or referer leak

Attacker replays UUID to an unprotected endpoint

~~SECURITY BY OBSCURITY~~

The UUID Fallacy

Hard-to-guess identifiers prevent enumeration, but do not replace access checks.

Leaked GUIDs instantly compromise unauthorized endpoints.

The Deterministic Fix

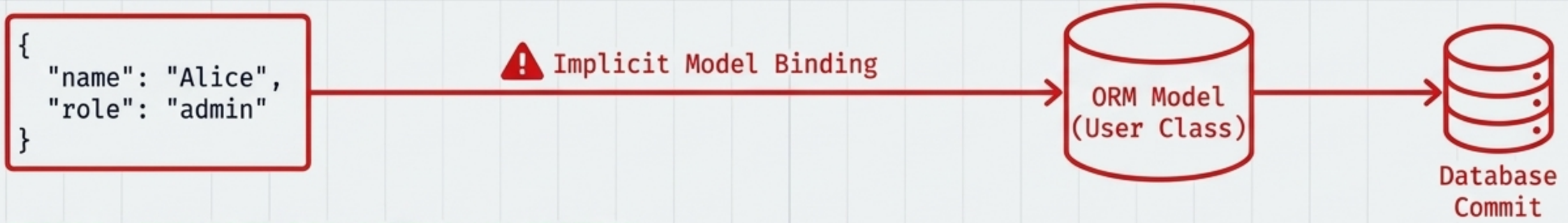
Move from scattered controller IF checks to centralized policy engines.

Utilize PostgreSQL Row-Level Security (RLS) or repository-scoped tenant queries unowned rows literally do not exist to the requesting user's database session.

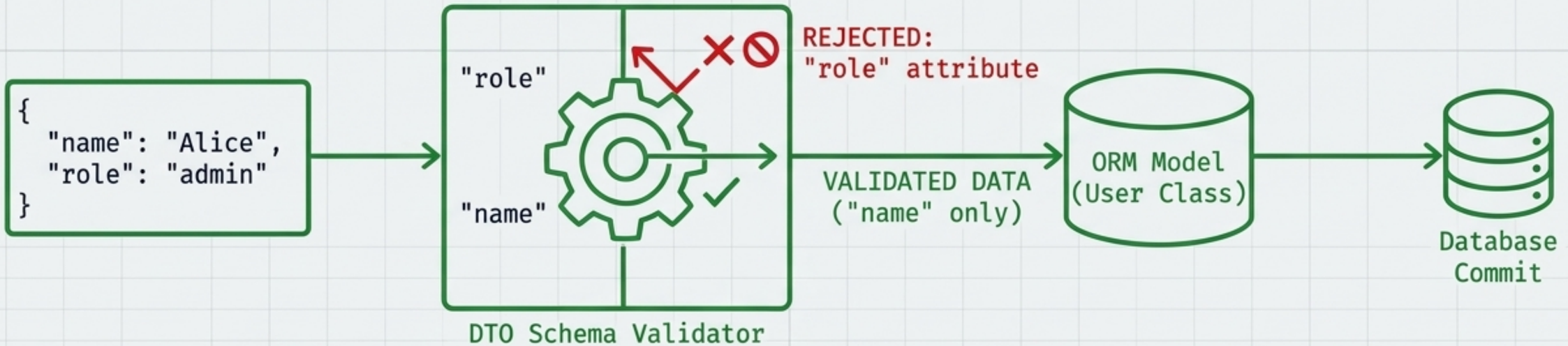
Gemini Notebook

MASS ASSIGNMENT (BOPLA): ARCHITECTURE COMPARISON

INSECURE: IMPLICIT MODEL BINDING



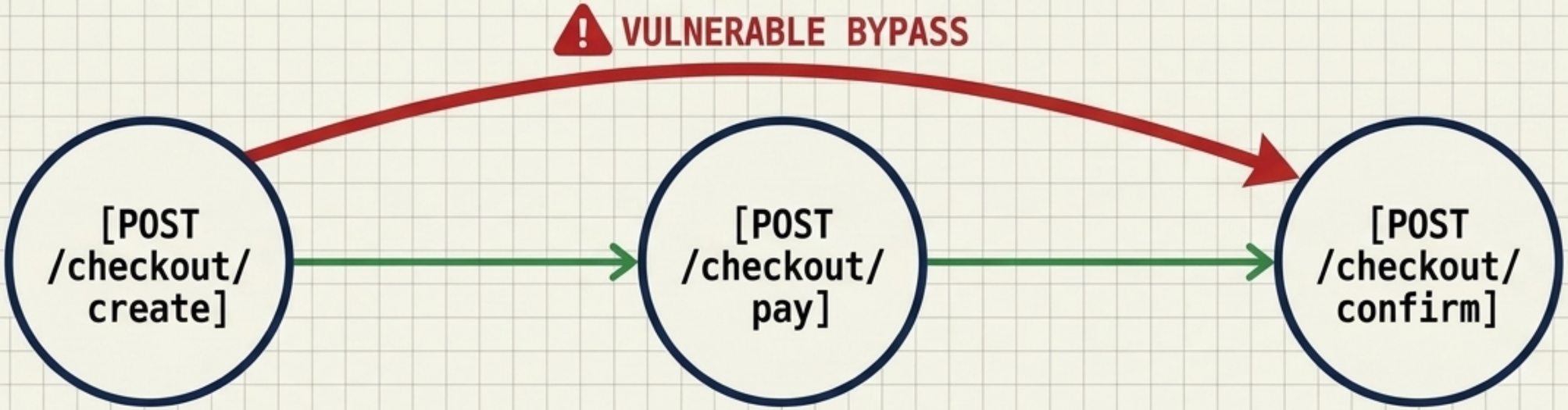
SECURE: EXPLICIT DTO SCHEMA VALIDATION



The Vulnerability:
Frameworks that implicitly bind HTTP request parameters to internal object variables blindly overwrite sensitive fields.

The Defense: Explicit Schema Validation.
Use strict Data Transfer Objects (DTOs) for inbound and outbound data. Reject unexpected attributes. Never reuse the same class for API payloads and database entities.

Gemini Notebook



The Attack Vector

Skipping required workflow steps by calling later-stage endpoints directly.

The Frontend Illusion

Because each endpoint is individually authenticated, traditional access control passes.

The backend incorrectly trusts that the frontend UI enforced the sequence.

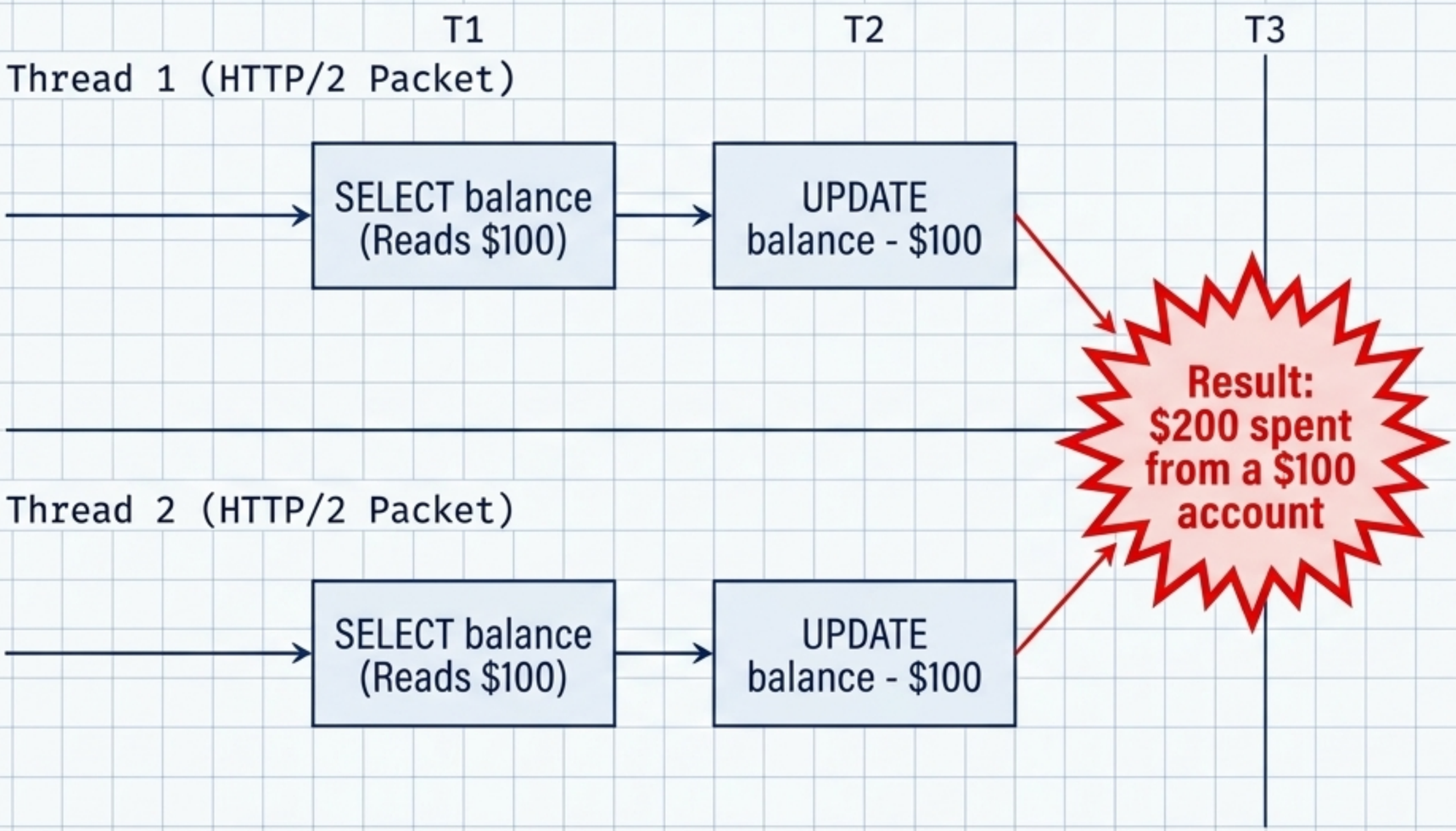
Prevention Guidance

Model workflows explicitly using finite state machines on the server.

Bind transition tokens to specific workflow stages. Reject invalid state transitions.

Gemini Notebook

SUB-MILLISECOND CONCURRENCY ATTACKS: HTTP/2 TIMING DIAGRAM



DETERMINISTIC FIXES

1. Atomic Conditional SQL ✅

```
UPDATE ...
 WHERE used = false
 AND balance >= 100
```

2. Pessimistic Locking ✅

`SELECT ... FOR UPDATE` (forces sequential processing)

3. Optimistic Locking ✅

`SERIALIZABLE` isolation with application-level retry logic

Last-byte HTTP/2 frame synchronization lands 20–50 requests at the database with sub-millisecond jitter, overwhelming default READ COMMITTED SQL isolation.

Gemini Notebook

MODERN API ABUSE VECTORS

Content-Type & Parser Vulnerabilities

HTTP POST Request

```
POST /api/update_user HTTP/1.1
Host: api.example.com
...
Content-Type: application/xml
...
```

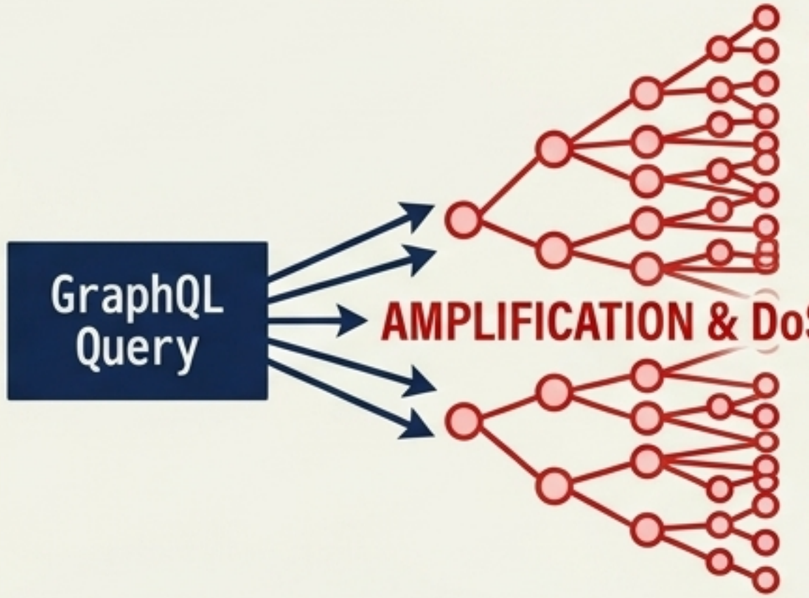
```
<!ENTITY xxe SYSTEM
"file:///etc/passwd">

<user><name>admin</name>
  <data>&xxe;</data>
</user>
```

Swapping expected JSON to XML forces the server to use an alternative, vulnerable parser.

Fix: Strictly validate **Content-Type** against an endpoint **allowlist** and drop mismatched payloads.

GraphQL Abuse Vectors



Introspection leaks expose schema.

Deeply nested circular queries trigger DoS.

Alias/batching enables rate-limit amplification.

Prevention Guidance

- ✅ Implement Query Cost Analysis ✅
- ✅ Persisted Queries ✅
- ✅ strict depth limits.

Gemini Notebook

JWT VULNERABILITY DISSECTION: HEADER & SIGNATURE FLAWS

HEADER (VULNERABLE)

```
{
  "alg": "none",
  "typ": "JWT"
}
```

THE alg:none ATTACK:
If the server derives the verification algorithm solely from the header, attackers strip the signature and bypass validation.

PAYLOAD (SECURE VALIDATION)

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022,
  "iss": "auth.example.com",
  "aud": "api.example.com",
  "exp": 1609459200,
  "jti": "aB12...c34"
}
```

MANDATORY VERIFICATION:
Verify Issuer ("iss"), Audience ("aud"), and Expiration ("exp"). Use "jti" (JWT ID) combined with server-side denylists for explicit logouts.

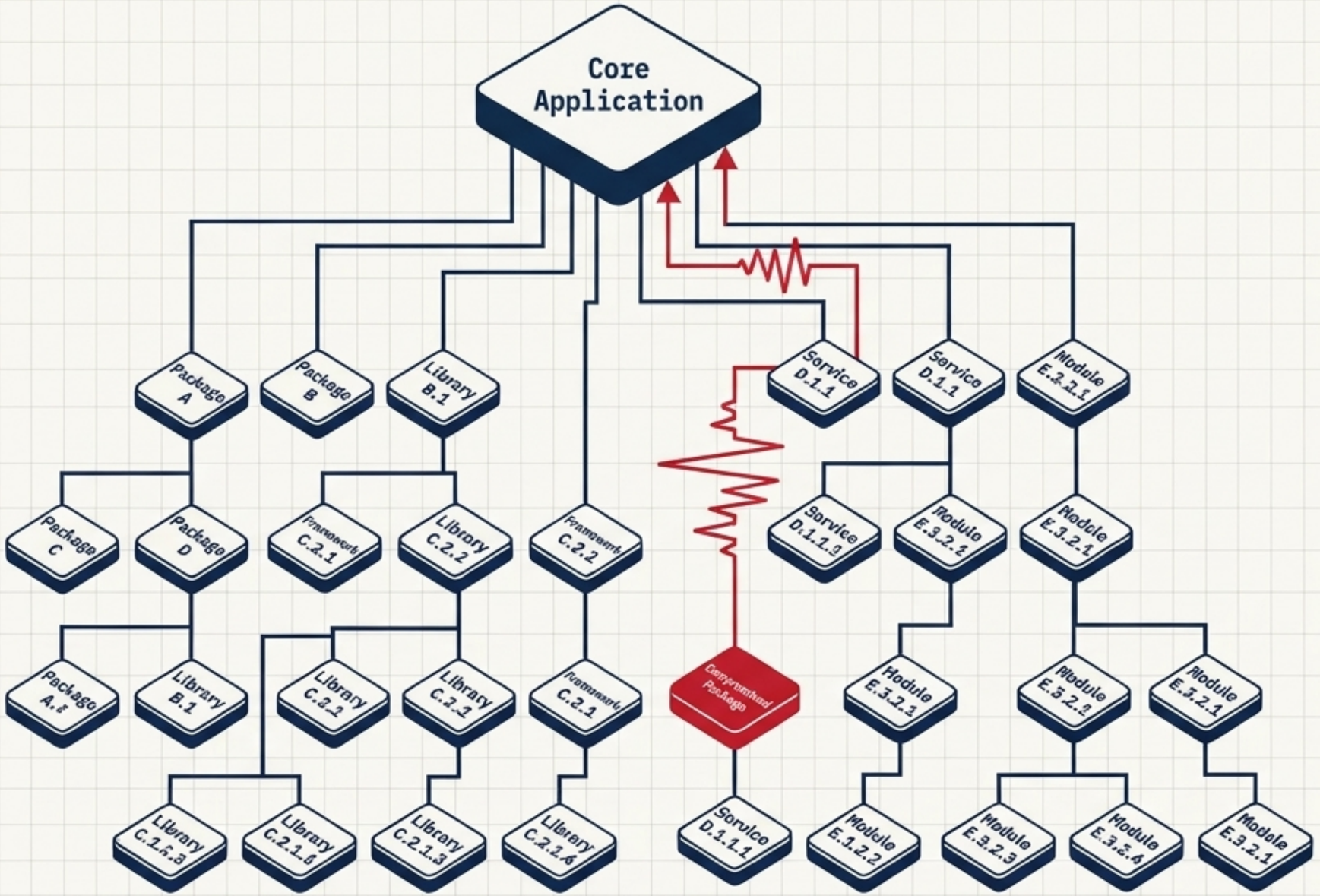
SIGNATURE (CONFUSION VECTOR)

MAC vs. SIGNATURE CONFUSION:
Attackers pass an RSA public key as an HMAC secret to forge tokens.

RSA Public Key (as HMAC Secret) → Used in HMAC Calculation → Forged Signature (Validates as HMAC)

OVERALL TAKEAWAY:
Relying parties must hardcode algorithm validation independently of the token header.

Gemini Notebook



MODERN SUPPLY CHAIN VECTORS

1. Typosquatting & Dependency Confusion: Tricking package managers into pulling malicious public packages.

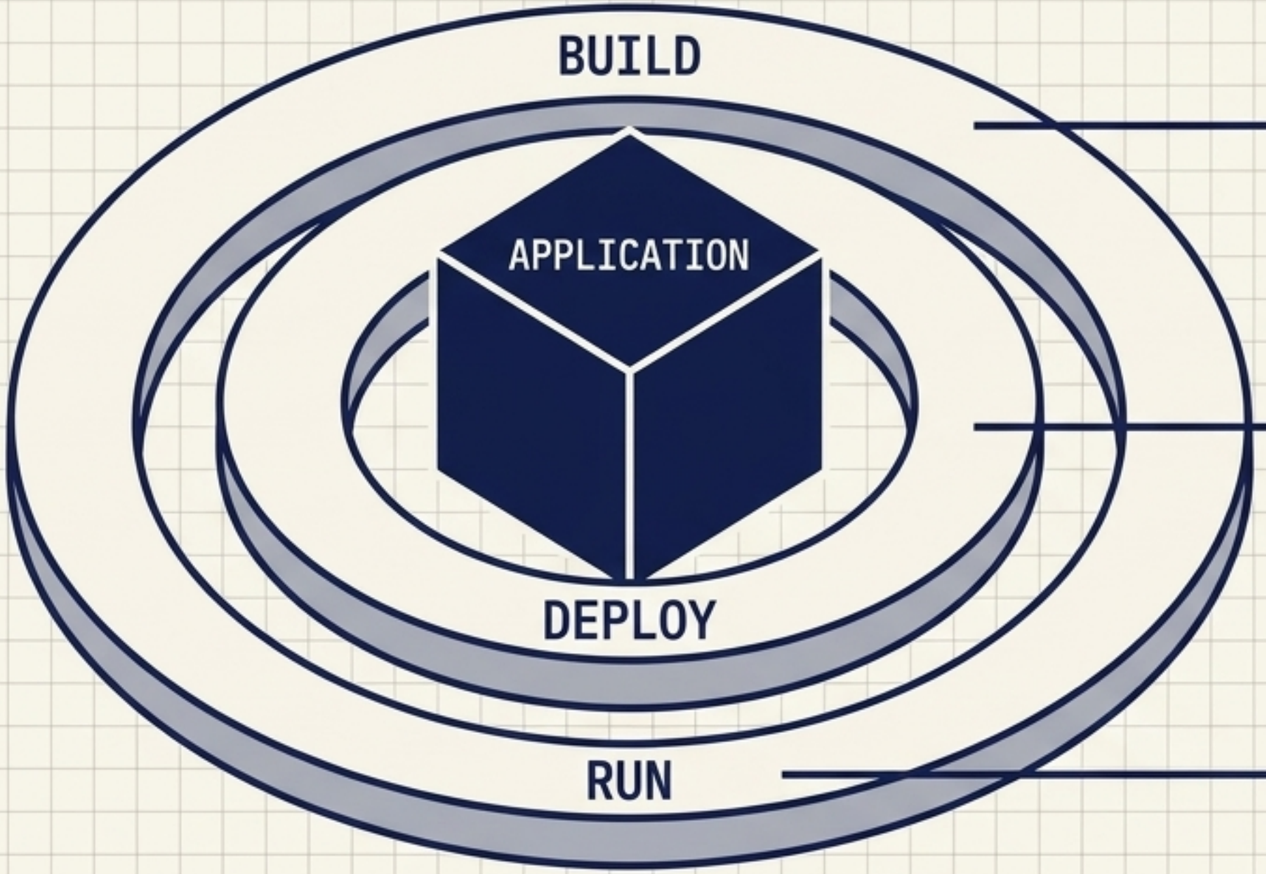
2. Malicious Hooks: npm/PyPI postinstall scripts stealing developer/CI credentials during build.

3. CI/CD Injection: Exploiting GitHub Actions pull_request_target to execute untrusted code.

Vulnerabilities in deep dependency graphs bypass all application-layer authorization checks.

Gemini Notebook

DEFENSE-IN-DEPTH: SECURING THE SOFTWARE SUPPLY CHAIN



Provenance Attestations (SLSA / Sigstore) to cryptographically verify build origins.

Deterministic Lockfiles. Pin exact hashes, not just versions, to prevent upstream tampering.

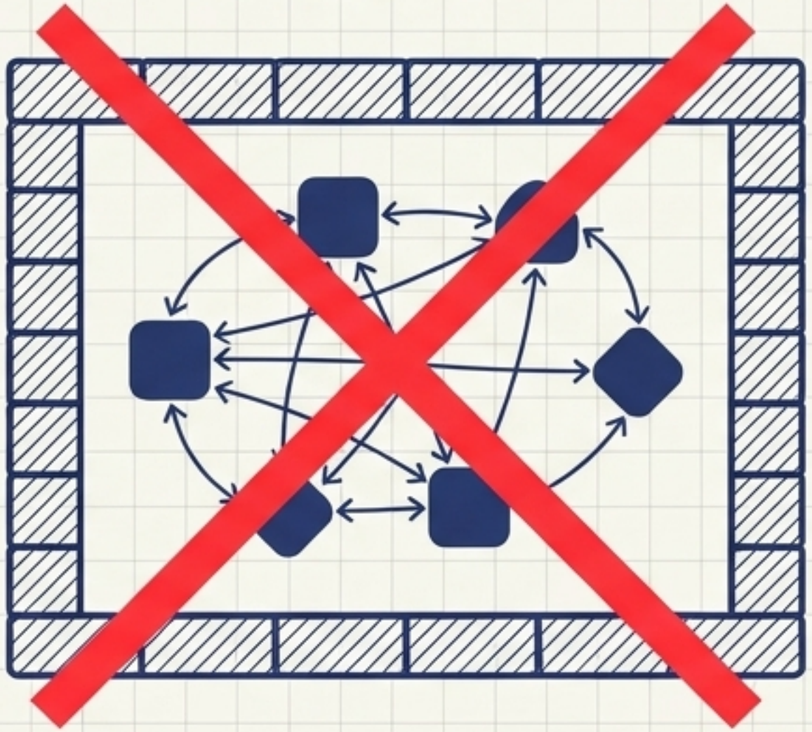
Runtime Blast-Radius Containment. Restrict microservice permissions.

A **compromised PDF rendering library** should not have **network access** to the **AWS Metadata service** or **user databases.**

Gemini Notebook

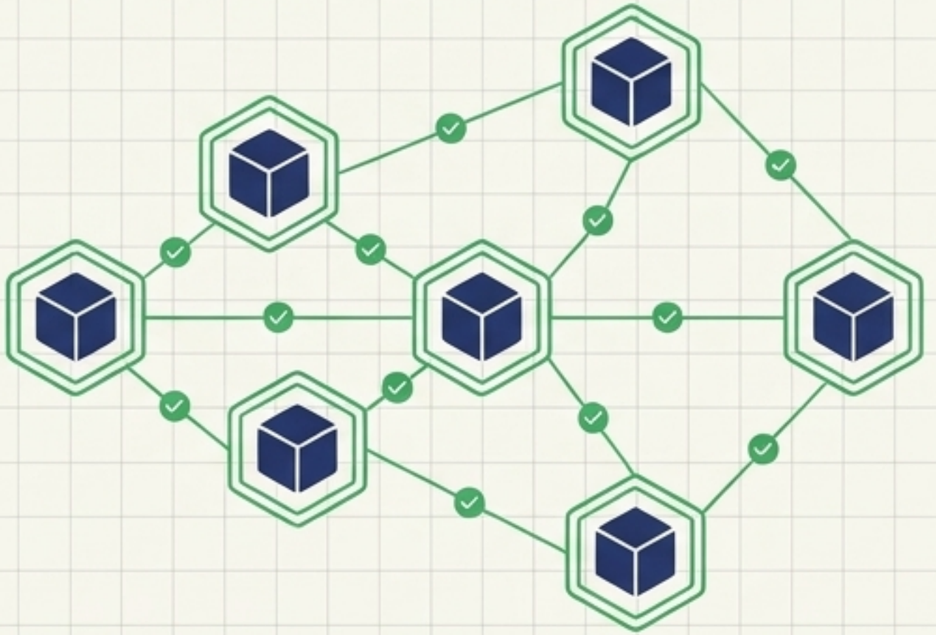
NIST SP 800-207: FROM PERIMETER SECURITY TO ZERO TRUST

OUTDATED



Implicit Trust (Internal VPC = Trusted)

MODERN



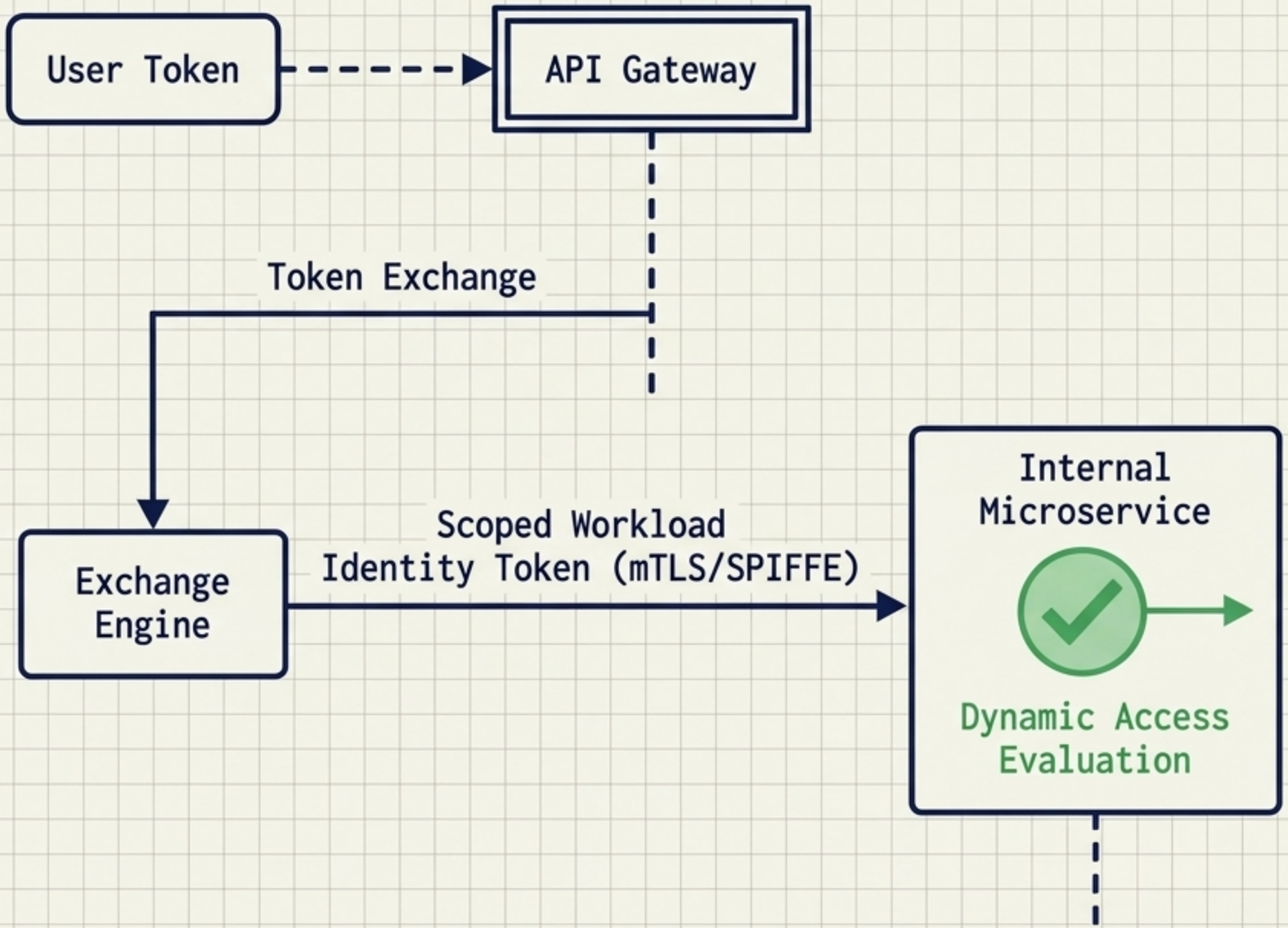
Micro-Perimeters: Secure Individual Resources & Workflows

NIST SP 800-207 CORE TENETS

1. **The Fall of the Perimeter:** Move defenses from static perimeters to individual users, assets, and resources.
2. **Zero Trust:** Assume no implicit trust is granted based solely on physical or network location.
3. **The Shift:** Protect resources and workflows, not network segments.

Gemini Notebook

DYNAMIC TRUST AND IDENTITY PROPAGATION: SEQUENCE DIAGRAM



CONTINUOUS EVALUATION

- Access is evaluated on every request, combining user identity with workload identity and environmental attributes.

AUDIT LOGGING

- Mandate structured, secret-rotated, token/PII-redacted audit logs across all micro-perimeters for continuous behavioral monitoring.

Gemini Notebook

API SECURITY HEADERS: CONFIGURATION CHECKLIST

CACHE-CONTROL: NO-STORE



Instructs private and shared caches not to store sensitive API responses (RFC 9111).

CONTENT-SECURITY-POLICY: FRAME-ANCESTORS ‘NONE’



Prevents clickjacking and UI redressing by explicitly blocking framing.

STRICT-TRANSPORT-SECURITY



Forces TLS connections, eliminating downgrade attacks.

X-CONTENT-TYPE-OPTIONS: NOSNIFF



Forces browsers to respect declared Content-Types, neutralizing MIME-sniffing script execution.

Gemini Notebook

SECURITY.TXT CONFIGURATION: VALIDATION & AUTHENTICATION

```
Contact: mailto:security@example.com
Encryption: https://example.com/pgp-key.txt
Canonical: https://example.com/.well-known/security.txt
Expires: 2025-12-31T23:59:59z
```

Standardized Disclosure (RFC 9116): Routes external vulnerability disclosures securely.

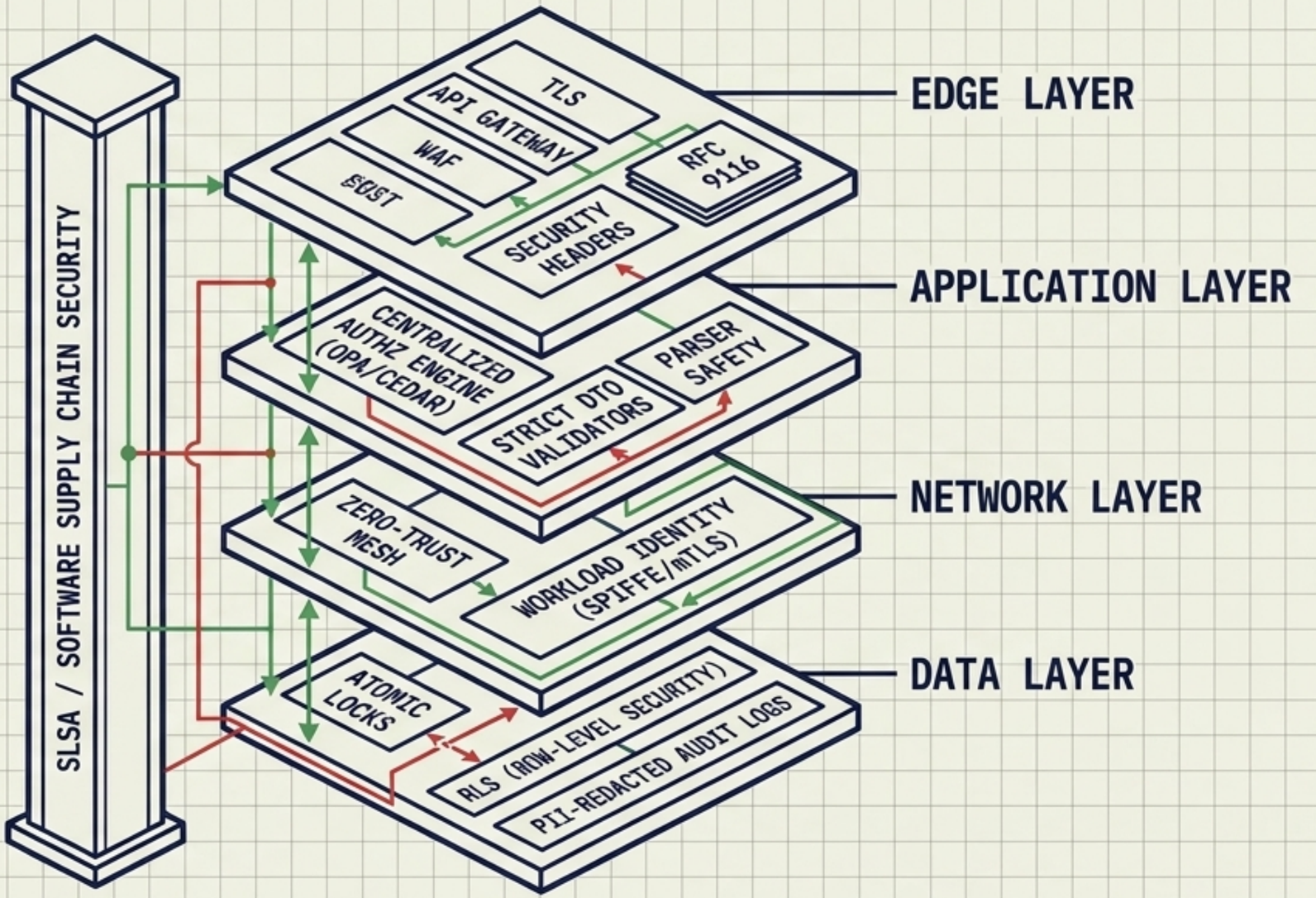
Canonical Trust: Always include Canonical URI to prevent attackers from hosting the file on compromised domains.

Digital Signatures: Sign the file with OpenPGP cleartext to authenticate location and protect data in transit.

```
-----BEGIN PGP SIGNATURE-----

HASH#A2E5HASHA2H6GAH31BANPGP5#MGDRCHHAESRASH
6MM#HMXALEWIIKEOTWRWAGHBdSHHHADANTRTARW5RING
HZXPWTLWENFDIMTBIW5#TONJARABKWKO0BYDND3AB7GF
ANN9XNAN8HMI08IHASHAAHAHAAN0HPHFSH6AIC5SHeGP
CGMeo:RNfftGEMKG4P2G#A@1b6N4BGGALTNDIoHBAAC#
###
-----END PGP SIGNATURE-----
```

Gemini Notebook



The Layered Defense-in-Depth Blueprint: Security is the synergistic overlap of micro-perimeters, deterministic logic, continuous identity evaluation, and cryptographically verified supply chains. Assume breach. Engineer for resilience.

Gemini Notebook