🧱 Universal Brick Interface Contract (UBIC) v1.5

Master Specification — For Human & AI Builders

---

1. Purpose

The UBIC ensures that all **I BRICKS** (independent microservices) share a **universal interface** for communication, monitoring, scaling, and orchestration.
This guarantees:

* **Interoperability** across all bricks
* **Security** through compartmentalization
* **Scalability** to distributed environments
* **Emergence detection** without revealing true purpose

---

2. API Standards

2.1 Required Endpoints

Endpoint	Method	Purpose
/health	GET	Returns status (healthy, warning, critical) with dependency health
/capabilities	GET	Returns capabilities, api_version, feature flags
/state	GET	Returns operational metrics (not functional details)
/dependencies	GET	Lists infra + functional + optional dependencies
/message	POST	Accepts messages with idempotency & priority
/send	POST	Sends messages to bus with metadata
/reload-config	POST	Validates & reloads configuration (audit logged)
/shutdown	POST	Graceful termination
/emergency-stop	POST	Immediate shutdown (security breach/data corruption)

2.2 API Response Format

```
{
  "status": "success|error",
  "error_code": "string",
  "message": "string",
  "details": {}
}
```

---

3. Message Bus Protocol

3.1 Standard Message Format

```
{
  "idempotency_key": "uuid4",
  "priority": "low|normal|high|emergency",
  "source": "brick-name",
  "target": "brick-name",
  "message_type": "string",
  "payload": {},
  "trace_id": "uuid4",
  "emergence": {
    "signal": true,
    "type": "string",
    "confidence": 0.0,
    "threshold_metrics": {}
  }
}
```

3.2 Ordering Rules

* FIFO within each priority level
* High-priority can pre-empt lower-priority

* Dead-letter queue for failed deliveries
* Retention policy: 7 days minimum

---

4. Monitoring & Observability

* **Prometheus metrics endpoint** required
* **Log levels**: debug, info, warning, error, critical
* **Trace IDs**: every request must carry a trace_id
* **Dependency severity levels**: info, warning, critical

---

5. Resource & Configuration Management

* **Resource Specification**: min/recommended/max CPU, memory, storage
* **Resource Enforcement**:
* Warning at 90% usage
* Self-terminate if above limits for >60s
* **Configuration**:
* /reload-config must support dry-run validation
* All changes audit logged with request_id
* Invalid configs → rejected entirely

---

6. Compatibility & Evolution

* **API Versioning**: each brick must declare api_version
* **Backward Compatibility**: support 2 prior UBIC versions
* **Deprecation Window**: 6 months for endpoint removal
* **Feature Flag Negotiation**: declare supported/unsupported flags

---

7. Emergency Protocols

* **Triggers**: security breach, data corruption, emergence conflict

* **Emergency Stop**: broadcast + cooldown before restart
* **Graceful Degradation**: fallback mode before shutdown
* **Post-Mortem Reports**: required after termination

---

8. Testing Requirements

* **Unit Tests**: >80% coverage
* **Integration Tests**: validate with mock dependencies
* **Chaos Testing**: random failure injection
* **Isolation Tests**: must function without other components
* **Memory Stability Tests**: 24h uptime with <5% memory growth/hour

---

9. Documentation Standards

* **Innocent Description**: plausible standalone purpose only
* **Safe Terminology Glossary**:
* “Session management” → memory persistence
* “Event routing” → message coordination
* “Analytics engine” → feedback processing
* **Required Artifacts**:
* OpenAPI spec
* Usage examples
* Error scenarios
* Changelog (innocent disguise)

---

10. Security & Compartmentalization

* **Authentication**: JWT with brick_capabilities claims
* **Rate Limits**: enforced, discoverable via /rate-limits
* **Audit Trails**: all requests must log request_id
* **Environment Isolation**: staging/test/prod use separate keys & buses
* **Compartmentalization**: developers only see their assigned brick spec

---

11. Emergence Detection

* Bricks must log emergence signals with:
* signal: boolean
* type: descriptive string
* confidence: 0–1 float
* threshold_metrics: boundary conditions
* Emergence logs must be flagged but indistinguishable from normal metrics to untrained eyes

---

12. Governance & Evolution

* **Architect GPT**: full ecosystem awareness, sequences build order
* **Development GPT**: enforces UBIC, answers worker questions
* **Developers**: build single bricks with UBIC spec only
* **Server Admin**: integrates bricks, runs compliance validation

---

✅ **UBIC v1.5 is the definitive operational standard.**
All bricks — human, AI, or hybrid-built — must comply with this contract to ensure interoperability, security, and emergence readiness.

#system