

ArgBMS — Server Hardware Sizing & Architecture

v4.1 | Indicative planning guide | Layout repair and source clarification | 03 Oct 2026

1. Overview & Assumptions

- Workloads: React static site, Spring Boot API + WebSocket, PostgreSQL, optional Redis/RabbitMQ, optional SMPP.
- Goal: predictable capacity with separation of concerns and growth headroom.
- Preference: Linux for production (lower overhead); Windows supported with ~25% RAM buffer.

2. Hardware Tiers

Tier	Target Load	CPU	RAM	Storage	Network	Notes
T1 — Small	20 users / ~300 devices	4 vCPU	16 GB	NVMe 500 GB	1 GbE	Mono-node (App + DB + Front).
T2 — Standard	100 users / 2–3k devices	8 vCPU (App & DB split)	32 GB	App: NVMe 500 GB; DB: NVMe 1 TB (RAID1/10)	1 GbE (VLAN for DB)	Three nodes: App/Front, DB, Cache/Queue.
T3 — Enterprise/HA	500 users / 10–15k devices	App: 8–16 vCPU; DB: 16 vCPU	App: 32–64 GB; DB: 64 GB	DB: 2×1 TB NVMe RAID1 + separate WAL volume	10 GbE App DB (or bonded 2×1 GbE)	LB×2, App×2, DB Primary+Standby, Cache/Queue×2.

Capacity (indicative): T1 300 WS / 30–60 API RPS; T2 2–3k WS / 150–300 API RPS; T3 10–15k WS / 600–1200 API RPS (workload-dependent).

Indicative planning guide - source discrepancy requires project sizing

T1 differs in the original v4.0 source: Hardware Tiers (page 1) = 4 vCPU / 16 GB RAM; Core Compute BOM (page 2) = 8 vCPU / 32 GB RAM. Both original examples are preserved. Confirm the intended configuration and test the project workload before procurement or production use.

v4.1 repairs layout and clarifies this discrepancy; it does not reconcile the examples or validate capacity. All capacity figures remain indicative and workload-dependent. Content basis: v4.0 (PDF metadata dated 14 September 2025).

Indicative planning guide - source discrepancy requires project sizing

T1 differs in the original v4.0 source: Hardware Tiers (page 1) = 4 vCPU / 16 GB RAM; Core Compute BOM (page 2) = 8 vCPU / 32 GB RAM. Both original examples are preserved. Confirm the intended configuration and test the project workload before procurement or production use.

3. Bill of Materials — Part A (Core Compute)

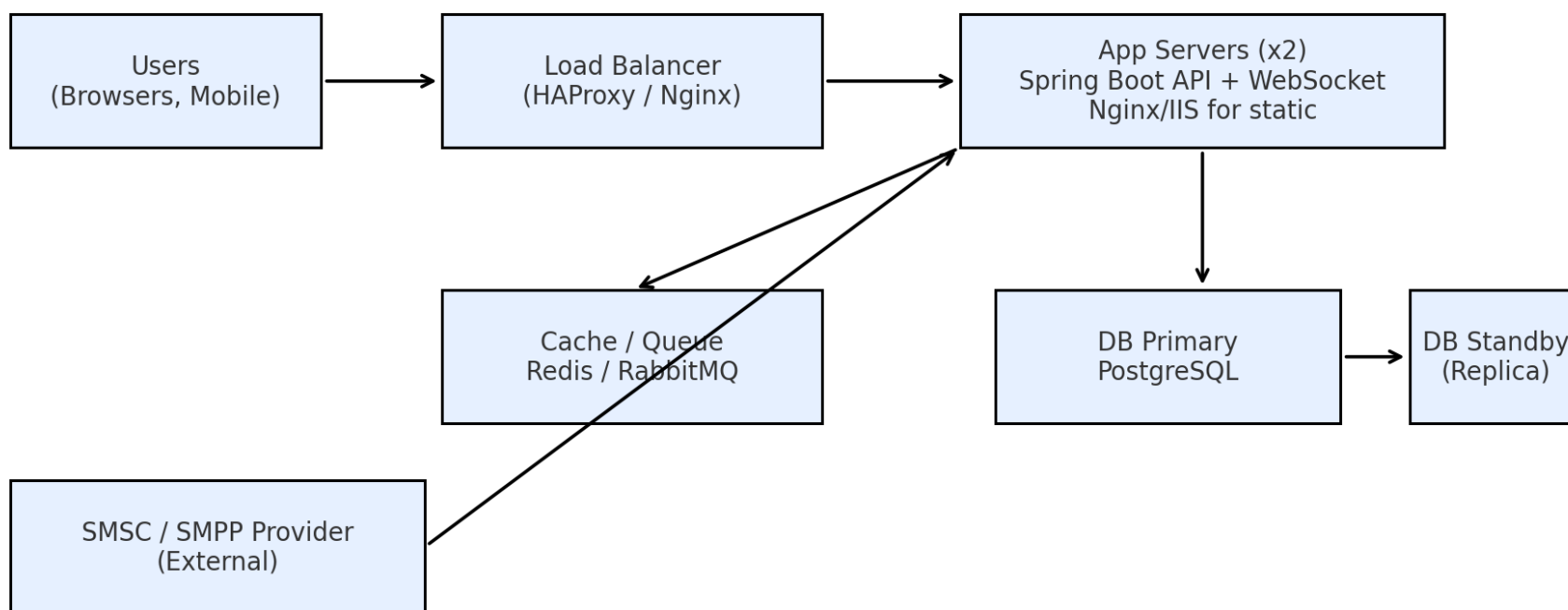
Layer	T1 (Single Node)	T2 (Three Nodes)	T3 (HA Cluster)
Front / LB	— (on App)	Nginx/HAProxy on App node	2× LB nodes: 4 vCPU, 8–16 GB RAM
App	8 vCPU / 32 GB, NVMe 500 GB	8 vCPU / 32 GB, NVMe 500 GB	2× nodes: 12 vCPU / 48 GB, NVMe 500 GB
DB	Same node (NVMe 500 GB)	8 vCPU / 32 GB, NVMe 1 TB (RAID1/10)	Primary: 16 vCPU / 64 GB, 2×1 TB NVMe RAID1 + WAL
DB Replica	—	Optional standby	1× Standby (same class as Primary)

3. Bill of Materials — Part B (Network / Cache / Power)

Layer	T1 (Single Node)	T2 (Three Nodes)	T3 (HA Cluster)
Cache / Queue	Optional (same node)	1× node: 4 vCPU / 8–16 GB RAM	2× nodes: 4–8 vCPU / 16 GB RAM
Network	1 GbE	1 GbE + VLAN for DB	10 GbE App DB (or bonded 1 GbE)
Power	UPS 1–1.5 kVA	UPS 1.5–2 kVA per node	UPS 2–3 kVA per node + dual PSUs

4. Reference Architecture Diagram

Traffic: Users → LB → App (API + WebSocket). App uses Redis/RabbitMQ for cache/queue and PostgreSQL for persistence. DB replicates to a standby. SMPP (external SMSC) connects from App.



Solid arrows = primary flows (HTTPS, WS, DB, Queue). SMPP shown for SMSC integrations.

Note: Diagram is logical; adapt network/security per environment (VLANs/ACLs).