



PRITT INVESTMENT PARTNERS

How Optimal Engine, BusinessOS, local agents, and client-owned cloud fit together.

A practical implementation map for the delivery team before architecture discovery and departmental module design.

THREE DISTINCT LAYERS

BusinessOS is the interface. Optimal Engine is the governed operating runtime. Pritt's infrastructure hosts the shared services.

LAYER	PRIMARY RESPONSIBILITY	WHAT THE TEAM CONFIGURES
BusinessOS User operating surface	Workspace switching, department modules, people and role views, tasks, knowledge, projects, communications, desktop tools, and agent actions.	Which modules appear, how they are laid out, which role sees them, and what actions they expose.
Optimal Engine Context and memory runtime	Organization and workspace topology, source intake, signal classification, claims, facts, memories, retrieval, context packages, workflows, skills, audit, and policy.	Pritt topology, node types, routing, glossary, memory policy, retrieval scope, workflow rules, and connectors.
Local providers User-side execution	Local models, CLI agents, desktop tools, files, terminals, and user-controlled credentials.	Supported models, agent commands, soul files, tool permissions, parameters, and fallback behavior.
Pritt cloud Shared infrastructure	Team synchronization, hosted integrations, shared databases, managed services, network controls, backups, and production observability.	Provider, accounts, regions, networking, identities, secrets, retention, backups, and service ownership.

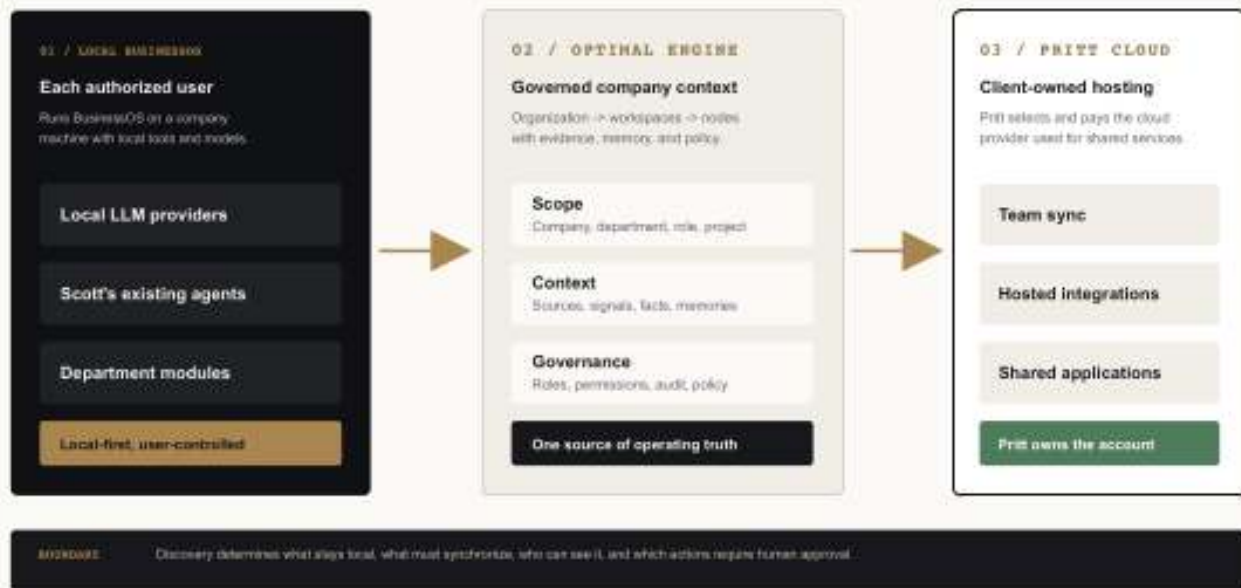
Do not collapse these layers into one vague “AI system.” Each has a separate owner, data boundary, failure mode, and acceptance test.

LOCAL FIRST, CLOUD WHEN SHARED

The design begins on the user's machine and introduces cloud infrastructure only where team operation requires it.

REFERENCE ARCHITECTURE / CLIENT-OWNED INFRASTRUCTURE

Local execution where it fits, shared cloud where the team needs it



Discovery question: for every tool, model, memory object, integration, and action, decide whether it is local-only, team-shared, cloud-hosted, or prohibited.

PROFIT: TECHNICAL IMPLEMENTATION ISSUE

ORGANIZATION -> WORKSPACE -> NODE

The org chart is not documentation after the fact. It becomes the routing and authorization model.

ORGANIZATION	WORKSPACE	NODE
Pritt operating boundary - Legal entities and companies. - Shared strategy and policies. - Identity and tenant administration. - Cross-workspace authority.	Department boundary - Operations, Marketing, Finance, and others. - Scoped modules and memories. - Department members and roles. - Workflows and connectors.	The things being operated - People, teams, products, projects. - Customers, systems, and processes. - Decisions, tasks, and evidence. - Agent and context objects.



Rule: never use transcript names, job titles, or guessed permissions as production truth. Pritt must confirm the org chart, role definitions, information boundaries, and approval rights.

REUSE BEFORE REPLACEMENT

Scott's current agents and soul files become discovery inputs, not automatic production assets.

ARTIFACT	REVIEW	IMPLEMENTATION DECISION
Soul files Identity and behavior	Purpose, role, tone, authority, restrictions, assumptions, stale instructions, and conflicts.	Keep, rewrite, split, or retire. Bind to an explicit role and workspace.
Setup commands Runtime boot	Dependencies, models, tools, environment variables, paths, security, and portability.	Package into a supported local setup with documented prerequisites and health checks.
Agent tools Action surface	Read and write scope, credentials, external side effects, retries, audit, and human approval.	Expose only approved tools through scoped permissions and inspectable execution.
Prompts and workflows Operating logic	Inputs, outputs, branching, errors, examples, data dependencies, and evaluation criteria.	Convert repeatable work into governed workflows or skill packages.
Memory and files Context sources	Provenance, freshness, confidentiality, duplication, contradictions, and ownership.	Ingest as sources, extract claims, confirm facts, and scope retrieval by workspace and role.

The objective is not to preserve every existing agent exactly. It is to preserve useful intent while making behavior supportable, observable, and safe for employees.

REQUIRED BEFORE ARCHITECTURE LOCK

The next technical session should produce evidence, owners, and decisions, not another high-level demo.

BRING FROM PRITT

Current-state package

- Complete org chart and legal entities.
- Selected department and department owner.
- Existing system diagram and data inventory.
- Agent commands, soul files, and representative runs.
- Cloud preference and security constraints.
- Named pilot users and acceptance authority.

PRODUCE TOGETHER

Architecture record

- Confirmed workspace and role topology.
- Local versus cloud placement matrix.
- One end-to-end pilot workflow.
- Integration and credential map.
- Risks, exclusions, and human gates.
- Test plan and acceptance criteria.

LOCK BEFORE BUILD

Delivery contract

- Scope and change control.
- Owners and decision cadence.
- Environments and access.
- Deployment and rollback.
- Support boundary.
- Expansion decision.

Technical baseline: the open-source Optimal Engine repo is the reference for the governed engine layer. The deployed Pritt solution also depends on BusinessOS, configured modules, Pritt-owned infrastructure, and implementation work.

No production architecture should be finalized until the team can trace one real item from source, through role-aware context, to an authorized human or agent outcome.