YTEMPIRE Technical Specifications

Version 1.0 - Local Deployment Edition

Table of Contents

Executive Summary

This technical specification document defines the infrastructure design and performance requirements for YTEMPIRE's local deployment, with a clear migration path to cloud infrastructure. The specifications are optimized for the target hardware (AMD Ryzen 9 7950X3D, RTX 5090, 128GB RAM) while maintaining enterprise-grade reliability and scalability patterns.

Key Design Principles:

- **Local-First Architecture**: Maximizing on-premises hardware utilization
- **Cloud-Ready Design**: Seamless migration path when scaling beyond local capacity
- **Performance Optimization**: Leveraging GPU acceleration and multi-core processing
- **Security by Design**: Defense-in-depth approach with isolated security zones
- **Operational Excellence**: Comprehensive monitoring and automated recovery

---

Infrastructure Design

Local Architecture Foundation

Primary Infrastructure Components

yaml

```yaml
infrastructure:
  compute:
    primary_workstation:
      cpu: AMD Ryzen 9 7950X3D
      cores: 16
      threads: 32
      base_clock: 4.2 GHz
      boost_clock: 5.7 GHz
      l3_cache: 128MB (3D V-Cache)

    gpu:
      model: NVIDIA RTX 5090
      vram: 32GB GDDR7
      cuda_cores: 16384
      tensor_cores: 512
      rt_cores: 128
      memory_bandwidth: 1TB/s

    memory:
      capacity: 128GB DDR5
      speed: 5600MHz
      channels: 2
      configuration: 4x32GB

    storage:
      primary:
        type: NVMe Gen5 SSD
        capacity: 4TB
        read_speed: 12GB/s
        write_speed: 10GB/s
        iops: 2M
      secondary:
        type: NVMe Gen4 SSD
        capacity: 8TB
        read_speed: 7GB/s
        write_speed: 6GB/s
      backup:
        type: NAS (Synology DS1821+)
        capacity: 48TB (6x8TB RAID6)
        network: 10GbE
```

Resource Allocation Strategy

python

```python
class ResourceAllocator:
    """Intelligent resource allocation for local deployment"""

    def __init__(self):
        self.cpu_allocation = {
            'system_reserved': 2,  # cores
            'orchestrator': 2,
            'trend_analysis': 4,
            'content_generation': 3,
            'video_processing': 3,
            'auxiliary_services': 2
        }

        self.memory_allocation = {
            'system_reserved': 16,  # GB
            'docker_services': 32,
            'ml_models': 40,
            'video_processing': 24,
            'cache_layer': 16
        }

        self.gpu_allocation = {
            'stable_diffusion': 8,  # GB VRAM
            'llm_inference': 12,
            'video_encoding': 6,
            'tensor_operations': 4,
            'system_reserved': 2
        }
```

Local-First Architecture Decisions

Component	Technology Choice	Justification
Container Runtime	Docker 24.0 + Docker Compose	Simplicity for local deployment, easy migration to K8s
Service Mesh	None (initially)	Overhead not justified for 6 services
Load Balancer	Nginx 1.24	Lightweight, powerful reverse proxy
Service Discovery	Docker DNS + Consul	Automatic with migration path
Secrets Management	HashiCorp Vault (	