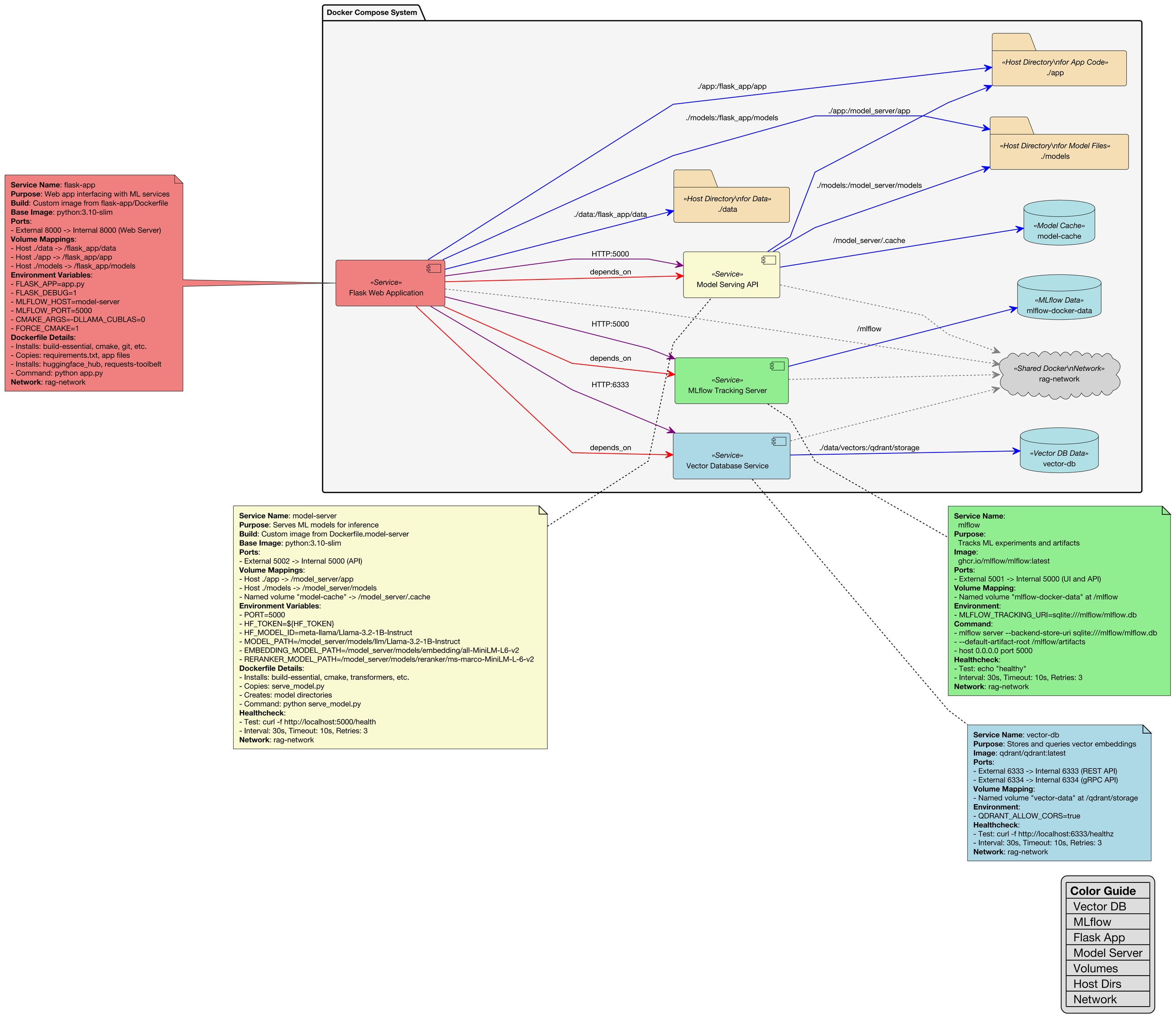
Docker Compose System
«Host Directory\nfor App Code»
./app
./app:/flask_app/app
./models:/flask_app/models
./app:/model_server/app
«Host Directory\nfor Model Files»
./models
«Host Directory\nfor Data»
./data
./data:/flask_app/data
./models:/model_server/models
«Model Cache»
model-cache
/model_server/.cache
«Service»
Flask Web Application
HTTP:5000
depends_on
«Service»
Model Serving API
«MLflow Data»
mlflow-docker-data
HTTP:5000
/mlflow
depends_on
«Shared Docker\nNetwork»
rag-network
«Service»
MLflow Tracking Server
HTTP:6333
depends_on
«Service»
Vector Database Service
./data/vectors:/qdrant/storage
«Vector DB Data»
vector-db
Service Name: flask-app
Purpose: Web app interfacing with ML services
Build: Custom image from flask-app/Dockerfile
Base Image: python:3.10-slim
Ports:
- External 8000 -> Internal 8000 (Web Server)
Volume Mappings:
- Host ./data -> /flask_app/data
- Host ./app -> /flask_app/app
- Host ./models -> /flask_app/models
Environment Variables:
- FLASK_APP=app.py
- FLASK_DEBUG=1
- MLFLOW_HOST=model-server
- MLFLOW_PORT=5000
- CMAKE_ARGS=-DLLAMA_CUBLAS=0
- FORCE_CMAKE=1
Dockerfile Details:
- Installs: build-essential, cmake, git, etc.
- Copies: requirements.txt, app files
- Installs: huggingface_hub, requests-toolbelt
- Command: python app.py
Network: rag-network
Service Name: model-server
Purpose: Serves ML models for inference
Build: Custom image from Dockerfile.model-server
Base Image: python:3.10-slim
Ports:
- External 5002 -> Internal 5000 (API)
Volume Mappings:
- Host ./app -> /model_server/app
- Host ./models -> /model_server/models
- Named volume "model-cache" -> /model_server/.cache
Environment Variables:
- PORT=5000
- HF_TOKEN=${HF_TOKEN}
- HF_MODEL_ID=meta-llama/Llama-3.2-1B-Instruct
- MODEL_PATH=/model_server/models/llm/Llama-3.2-1B-Instruct
- EMBEDDING_MODEL_PATH=/model_server/models/embedding/all-MiniLM-L6-v2
- RERANKER_MODEL_PATH=/model_server/models/reranker/ms-marco-MiniLM-L-6-v2
Dockerfile Details:
- Installs: build-essential, cmake, transformers, etc.
- Copies: serve_model.py
- Creates: model directories
- Command: python serve_model.py
Healthcheck:
- Test: curl -f http://localhost:5000/health
- Interval: 30s, Timeout: 10s, Retries: 3
Network: rag-network
Service Name:
mlflow
Purpose:
Tracks ML experiments and artifacts
Image:
ghcr.io/mlflow/mlflow:latest
Ports:
- External 5001 -> Internal 5000 (UI and API)
Volume Mapping:
- Named volume "mlflow-docker-data" at /mlflow
Environment:
- MLFLOW_TRACKING_URI=sqlite:///mlflow/mlflow.db
Command:
- mlflow server --backend-store-uri sqlite:///mlflow/mlflow.db
- --default-artifact-root /mlflow/artifacts
- host 0.0.0.0 port 5000
Healthcheck:
- Test: echo "healthy"
- Interval: 30s, Timeout: 10s, Retries: 3
Network: rag-network
Service Name: vector-db
Purpose: Stores and queries vector embeddings
Image: qdrant/qdrant:latest
Ports:
- External 6333 -> Internal 6333 (REST API)
- External 6334 -> Internal 6334 (gRPC API)
Volume Mapping:
- Named volume "vector-data" at /qdrant/storage
Environment:
- QDRANT_ALLOW_CORS=true
Healthcheck:
- Test: curl -f http://localhost:6333/healthz
- Interval: 30s, Timeout: 10s, Retries: 3
Network: rag-network
Color Guide
Vector DB
MLflow
Flask App
Model Server
Volumes
Host Dirs
Network