TbD V6.0 Deployment Manifest

Status: Production Ready (Native Insight) **Architecture:** 4-Service Mesh (Dispatcher, Worker, Encoder, Detector)

1. Directory Structure

Ensure your local project follows this exact layout before running scripts.

```
tbd-v6/
├── deploy_v6_master.ps1       # [Script] Deploys Service A & B
├── deploy_encoder.ps1         # [Script] Deploys Service C
├── deploy_detector.ps1        # [Script] Deploys Service D
├── requirements.txt           # [Config] Worker Dependencies
├── Dockerfile                 # [Config] Worker Container
├── app/
│   ├── main.py
│   ├── schema.py
│   └── services/
│       ├── worker.py          # [Code] V6 Execution Logic
│       ├── pipeline.py
│       ├── dispatcher.py
│       └── ...
├── tbd-encoder/
│   ├── Dockerfile             # [Config] Encoder Container
│   ├── encoder_requirements.txt
│   ├── lstm_model.h5          # [Asset] Model Weights
│   ├── tokenizer.pickle       # [Asset] Tokenizer
│   └── encoder_app/
│       └── main.py            # [Code] Rebuild Architecture Logic
└── tbd-detector/
    ├── Dockerfile             # [Config] Detector Container
    ├── detector_requirements.txt
    ├── model_yolo.zip         # [Asset] Zipped SavedModel
    └── detector_app/
        └── main.py            # [Code] Native SavedModel Logic
```

2. Service C: Temporal Encoder (The Memory)

Strategy: Keras Architecture Rebuild (Bypasses Version Conflict).

tbd-encoder/Dockerfile

```
FROM python:3.10-slim
WORKDIR /usr/src/app
RUN apt-get update && apt-get install -y build-essential libhdf5-dev pkg-config && rm -rf
/var/lib/apt/lists/*
COPY encoder_requirements.txt .
RUN pip install --no-cache-dir -r encoder_requirements.txt
COPY tokenizer.pickle .
COPY lstm_model.h5 .
COPY encoder_app/ ./encoder_app/
ENV PORT=8080
CMD ["uvicorn", "encoder_app.main:app", "--host", "0.0.0.0", "--port", "8080"]
```

tbd-encoder/encoder_requirements.txt

```
fastapi==0.104.1
uvicorn==0.24.0
pydantic==2.5.2
tensorflow-cpu==2.15.0
h5py==3.10.0
numpy>=1.26.0,<2.0.0
```

deploy_encoder.ps1

```
$PROJECT_ID = "tbd-v2"
$REGION = "us-central1"
$SERVICE_NAME = "tbd-temporal-encoder"
$WORKER_SA = "tbd-worker-sa@$PROJECT_ID.iam.gserviceaccount.com"

gcloud config set project $PROJECT_ID
gcloud builds submit tbd-encoder/ --tag
"$REGION-docker.pkg.dev/$PROJECT_ID/tbd-encoder-repo/temporal-encoder:v6-rebuild"
gcloud run deploy $SERVICE_NAME --image
"$REGION-docker.pkg.dev/$PROJECT_ID/tbd-encoder-repo/temporal-encoder:v6-rebuild"
--region $REGION --no-allow-unauthenticated --memory 2Gi
gcloud run services add-iam-policy-binding $SERVICE_NAME --region $REGION
--member="serviceAccount:$WORKER_SA" --role="roles/run.invoker"
Write-Host "Encoder Deployed: $(gcloud run services describe $SERVICE_NAME --region
$REGION --format 'value(status.url)')"
```

3. Service D: Object Detector (The Eyes)

Strategy: Native SavedModel (Zip) + TFSMLayer (Bypasses OS Incompatibility).

tbd-detector/Dockerfile

```
FROM python:3.10-slim
WORKDIR /usr/src/app
RUN apt-get update && apt-get install -y libgl1 libglib2.0-0 unzip && rm -rf /var/lib/apt/lists/*
COPY detector_requirements.txt .
RUN pip install --no-cache-dir -r detector_requirements.txt
COPY model_yolo.zip .
RUN unzip model_yolo.zip -d model_yolo_dir && rm model_yolo.zip
COPY detector_app/ ./detector_app/
ENV PORT=8080
CMD ["uvicorn", "detector_app.main:app", "--host", "0.0.0.0", "--port", "8080"]
```

tbd-detector/detector_requirements.txt

```
fastapi==0.104.1
uvicorn==0.24.0
pydantic==2.5.2
tensorflow-cpu==2.17.0
numpy>=1.26.0,<2.0.0
opencv-python-headless==4.8.1.78
python-multipart
```

deploy_detector.ps1

```
$PROJECT_ID = "tbd-v2"
$REGION = "us-central1"
$SERVICE_NAME = "tbd-object-detector"
$WORKER_SA = "tbd-worker-sa@$PROJECT_ID.iam.gserviceaccount.com"

if (-not (Test-Path "tbd-detector/model_yolo.zip")) { Write-Error "Missing model_yolo.zip"; exit 1 }

gcloud config set project $PROJECT_ID
gcloud builds submit tbd-detector/ --tag
"$REGION-docker.pkg.dev/$PROJECT_ID/tbd-detector-repo/object-detector:v6-native"
gcloud run deploy $SERVICE_NAME --image
"$REGION-docker.pkg.dev/$PROJECT_ID/tbd-detector-repo/object-detector:v6-native" --region
$REGION --no-allow-unauthenticated --memory 2Gi
gcloud run services add-iam-policy-binding $SERVICE_NAME --region $REGION
--member="serviceAccount:$WORKER_SA" --role="roles/run.invoker"
```

```
Write-Host "Detector Deployed: $(gcloud run services describe $SERVICE_NAME --region
$REGION --format 'value(status.url)')"
```

4. Service B: The Worker (The Brain)

Strategy: Full Integration (Removes Bypass Flags).

`Dockerfile` (Root)

```
FROM python:3.10-slim
WORKDIR /usr/src/app
ENV PYTHONUNBUFFERED=True
RUN apt-get update && apt-get install -y tesseract-ocr libtesseract-dev libgl1 libglib2.0-0 ffmpeg
&& rm -rf /var/lib/apt/lists/*
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY ./app /usr/src/app/app
ENV PORT=8080
CMD sh -c "uvicorn app.main:app --host 0.0.0.0 --port \${PORT}"
```

`app/services/worker.py`

(Ensure this file contains the V6 logic with `_enrich_with_temporal_context` and `_fetch_iot_telemetry` functions provided in Phase 3 chat).

`deploy_v6_master.ps1`

```
$PROJECT_ID = "tbd-v2"
$REGION = "us-central1"
$WORKER_SERVICE = "tbd-worker"
$DISPATCHER_SERVICE = "tbd-dispatcher"
$WORKER_SA = "tbd-worker-sa@$PROJECT_ID.iam.gserviceaccount.com"

gcloud config set project $PROJECT_ID

# 1. Auto-Discovery
$ENCODER_URL = gcloud run services describe tbd-temporal-encoder --region $REGION
--format 'value(status.url)'
$DETECTOR_URL = gcloud run services describe tbd-object-detector --region $REGION
--format 'value(status.url)'
```

```
if (-not $ENCODER_URL -or -not $DETECTOR_URL) { Write-Error "Dependencies missing.
Deploy Encoder/Detector first."; exit 1 }

# 2. Build Core Image
gcloud builds submit . --tag
"$REGION-docker.pkg.dev/$PROJECT_ID/tbd-repo/tbd-v6-core:latest"

# 3. Deploy Worker (With Injected URLs)
gcloud run deploy $WORKER_SERVICE `
   --image "$REGION-docker.pkg.dev/$PROJECT_ID/tbd-repo/tbd-v6-core:latest" `
   --region $REGION `
   --service-account $WORKER_SA `
   --no-allow-unauthenticated `
   --memory 4Gi `
   --cpu 2 `
   --timeout 3600 `
   --set-env-vars
"SERVICE_TYPE=worker,GCP_PROJECT_ID=$PROJECT_ID,TEMPORAL_ENCODER_URL=
$ENCODER_URL,OBJECT_DETECTOR_URL=$DETECTOR_URL"

# 4. Deploy Dispatcher
gcloud run deploy $DISPATCHER_SERVICE `
   --image "$REGION-docker.pkg.dev/$PROJECT_ID/tbd-repo/tbd-v6-core:latest" `
   --region $REGION `
   --allow-unauthenticated `
   --set-env-vars "SERVICE_TYPE=dispatcher,GCP_PROJECT_ID=$PROJECT_ID"

# 5. Pub/Sub Link
$WORKER_URL = gcloud run services describe $WORKER_SERVICE --region $REGION
--format 'value(status.url)'
gcloud pubsub subscriptions update tbd-worker-sub --push-endpoint=$WORKER_URL

Write-Host "V6 MASTER DEPLOYMENT COMPLETE"
Write-Host "Dispatcher URL: $(gcloud run services describe $DISPATCHER_SERVICE --region
$REGION --format 'value(status.url)')"
```