

MaxTune-SD_TSD_Kapitel-2.3_v2.1 – Technical System Documentation

Containerization and Runtime Logic

Version: 2.1 Effective from: 25.04.2025 Status: 100/100 – Approved Note:
Compatibility with Max systems is based on documented interfaces, not fixed version bindings.

2.3 Containerization and Runtime Logic

MaxTune executes learning segments within isolated containers, each defined by a runtime envelope that encapsulates:

- Execution triggers
- Allowed input/output interfaces
- Temporal boundaries
- Policy constraints

Containerization ensures that every learning instance remains reproducible, inspectable, and policy-compliant—irrespective of the runtime environment.

The runtime logic is adaptive but constrained. Containers are not free agents; they act within a deterministic framework of:

- Runtime quotas
- MaxReg policy validation checkpoints
- Lifecycle restrictions and kill-switches

This architecture prevents drift, isolates errors, and ensures that every container can be shut down, paused, or forked without impact on global system integrity.

MaxTune does not merely run models. It runs controllable, traceable, and sovereign learning containers—making runtime behavior an auditable asset, not a risk.

Document Hash (SHA256):

e014a9713d8fbfbc573175585f108f416c3f85c49e14f4e2c81e8ae8414e4ba9