

# MLKit Core — Header Interface & Contract Specification

This document defines the abstract interfaces, method signatures, parameter types, and responsibilities for the core `modelkit` headers. It outlines the contract layer without concrete ML backend dependencies, strictly serving the Bring-Your-Own-Framework (BYOF) architecture.

---

## 1. Configuration Header: `modelkit/config.py`

**Role:** Manages project configuration parsing, directory path resolution, and hardware device discovery.

### Class: `BaseConfig`

- `__init__(config_path: Optional[Union[str, Path]] = None) -> None`
    - *Purpose:* Initializes project settings by parsing `mlkit.json` or fallback defaults.
    - *Parameters:* Optional file path to the project configuration manifest.
  - `resolve_device() -> str`
    - *Purpose:* Inspects available local hardware and resolves target devices ("`cuda`", "`mps`", or "`cpu`") when declared as "`auto`".
    - *Returns:* Standardized string identifier for the execution device.
  - `to_dict() -> Dict[str, Any]`
    - *Purpose:* Serializes the active configuration into a dictionary for experiment tracking and manifest exports.
    - *Returns:* Key-value mapping of current configuration parameters.
- 

## 2. Data Pillar Header: `modelkit/data.py`

**Role:** Standardizes dataset ingestion, schema validation, and partition contracts across tabular, vision, text, or streaming inputs.

### Class: `Dataset(ABC)`

- `__init__(name: str, source: Optional[Union[str, Path]] = None, config: Optional[BaseConfig] = None) -> None`
  - *Purpose:* Sets dataset naming metadata, file/URI anchors, and configuration binds.
- `load(source: Optional[Union[str, Path]] = None, **kwargs: Any) -> Any [Abstract]`
  - *Purpose:* Ingests raw data into memory, buffers, or generator references.
  - *Parameters:* Optional source URI/path and backend-specific ingestion flags.
  - *Returns:* Reference to the loaded internal dataset instance.

- **split(train: float = 0.8, validation: float = 0.1, test: float = 0.1, \*\*kwargs: Any) -> Tuple[Any, Any, Any] [Abstract]**
    - *Purpose:* Partitions the dataset into train, validation, and test subsets based on fractional ratios.
    - *Parameters:* Fractional split sizes summing to 1.0; optional random seeds and stratification rules.
    - *Returns:* Tuple of (train\_split, validation\_split, test\_split).
  - **validate() -> bool**
    - *Purpose:* Verifies schema integrity, null-value thresholds, or required column/feature presence.
    - *Returns:* Boolean indicating readiness for training or evaluation.
- 

### 3. Model Pillar Header: **modelkit/models.py**

**Role:** Defines model initialization, checkpoint persistence, forward inference, and standalone evaluation hooks.

#### **Class: Model(ABC)**

- **\_\_init\_\_(name: str, config: Optional[Dict[str, Any]] = None) -> None**
    - *Purpose:* Initializes model identifiers, hyperparameter configurations, and internal instance handles.
  - **predict(inputs: Any, \*\*kwargs: Any) -> Any [Abstract]**
    - *Purpose:* Executes forward inference across batch or single-instance inputs.
    - *Parameters:* Raw or transformed input payloads, tensors, arrays, or text strings.
    - *Returns:* Model predictions, logits, token streams, or class labels.
  - **save(destination: Union[str, Path], \*\*kwargs: Any) -> None [Abstract]**
    - *Purpose:* Serializes model parameters, checkpoints, and weights to disk.
    - *Parameters:* Target directory or file destination path.
  - **load(source: Union[str, Path], \*\*kwargs: Any) -> None [Abstract]**
    - *Purpose:* Restores weights from storage or connects to an external local runtime process (such as Ollama).
    - *Parameters:* Checkpoint directory, file path, or service connection string.
  - **evaluate(dataset: Any, \*\*kwargs: Any) -> Dict[str, float]**
    - *Purpose:* Optional baseline hook computing performance metrics directly on a target dataset.
    - *Returns:* Dictionary of metric keys mapped to numerical scores.
- 

### 4. Lifecycle Pillar: **modelkit/lifecycle.py**

**Role:** Coordinates the execution pipeline across training workflows, evaluation runs, checkpointing, and local serving.

## Class: **BaseTrainer(ABC)**

- **fit(model: Model, dataset: Dataset, \*\*kwargs: Any) -> Dict[str, Any]** *[Abstract]*
  - *Purpose:* Coordinates optimization steps, data consumption, and convergence tracking.
  - *Parameters:* Active **Model** and **Dataset** instances, plus optimizer/epoch keyword arguments.
  - *Returns:* Dictionary reporting training status, loss curves, and execution metrics.
- **checkpoint(model: Model, destination: Union[str, Path], step: Optional[int] = None) -> Path**
  - *Purpose:* Writes an experiment snapshot containing weights, config state, and metrics.
  - *Returns:* Path to the saved checkpoint directory.

## Class: **BaseEvaluator(ABC)**

- **evaluate(model: Model, dataset: Dataset, \*\*kwargs: Any) -> Dict[str, float]** *[Abstract]*
  - *Purpose:* Evaluates model quality against held-out splits (validation or test).
  - *Parameters:* Active **Model** instance and partitioned **Dataset**.
  - *Returns:* Structured dictionary of metric names mapped to numerical scores (e.g., accuracy, loss, F1).

## Class: **BaseInference(ABC)**

- **run(model: Model, raw\_input: Any, \*\*kwargs: Any) -> Any** *[Abstract]*
    - *Purpose:* Normalizes raw payloads, triggers model inference, and packages structured responses for local execution or serving endpoints.
    - *Parameters:* Unprocessed incoming input payload or request dictionary.
    - *Returns:* Formatted prediction output dictionary or list.
- 

## 5. Dynamic Discovery & Registry: **modelkit/registry.py**

**Role:** Binds string identifiers declared in **mlkit.json** to user subclasses dynamically.

### Functions

- **register(category: str, name: str) -> Callable[[Type[Any]], Type[Any]]**
  - *Purpose:* Decorator that registers a class under a functional domain (e.g., "model", "dataset", "trainer").
- **get(category: str, name: str) -> Type[Any]**

- *Purpose:* Looks up and returns the registered class reference by name.
- *Raises:* `KeyError` if the requested identifier has not been registered.