Component Specification

A **Component Specification** defines the metadata, container configuration, protocols, and behavioral contracts of a computational unit (component) that can be onboarded into the system. These components are indexed, made searchable and discoverable in the components registry, these components can be readily instantiated as blocks for serving.

This spec is processed by the `add_component_IR` function and saved into the `Component` collection in the database. Components can later be referenced in blocks, vDAGs, or management operations.

Top-Level Structure

The spec must be submitted under:

```
{
  "body": {
    "spec": {
      "values": {
        ...
      }
    }
  }
}
```

Fields

Field	Type	Required	Description
`componentId`	`object`	Yes	Structured identifier for the component. See breakdown below.
`componentId.name`	`string`	Yes	Component name.
`componentId.version`	`string`	Yes	Version of the component.
`componentId.releaseTag`	`string`	Yes	Release tag for the component version.

Field	Type	Required	Description
`componentURI`	`string`	Yes	Unique URI of the component. System-generated in most cases: `<componentType>.<name>:<version>-<relea`
`componentType`	`string`	Yes	Type/category of the component (e.g., model, service, utility).
`containerRegistryInfo`	`object`	No	Metadata about the container image associated with the component. See below.
`containerRegistryInfo.containerImage`	`string`	Yes (if present)	Full container image path.
`containerRegistryInfo.containerRegistryId`	`string`	Yes (if present)	Identifier of the registry the image is stored in.
`containerRegistryInfo.containerImageMetadata`	`object`	No	Metadata like description, author, etc.
`containerRegistryInfo.componentMode`	`string`	Yes (if present)	Enum: `"aios"` or `"third_party"` indicating the component's origin.
`componentMetadata`	`object`	No	Custom metadata about the component.

Field	Type	Required	Description
`componentInitData`	`object`	No	Initialization data or configuration required during component startup.
`componentInputProtocol`	`object`	No	Description or schema of accepted inputs.
`componentOutputProtocol`	`object`	No	Description or schema of generated outputs.
`policies`	`object`	No	Policies attached to the component (e.g., lifecycle, security, governance).
`componentManagementCommandsTemplate`	`object`	No	Defines supported management commands and their structures.
`componentInitSettings`	`object`	No	Runtime settings such as flags, modes, or tuning parameters.
`componentParameters`	`object`	No	Parameters used for processing/inference, typically at runtime.
`componentInitSettingsProtocol`	`object`	No	Describes the expected format and structure of init settings.

Field	Type	Required	Description
`componentInitParametersProtocol`	`object`	No	Describes the expected structure of run-time/inference parameters.
`tags`	`array`	No	List of searchable tags associated with the component.

Protocol Template Structure

Protocol templates define the **structure, type, constraints, and description** of data expected in a component's configuration, input, output, or parameters. These templates are **JSON schemas** that allow both human and machine interpretation of data contracts.

They are used in the following fields of a component specification:

- `componentInitSettingsProtocol`
- `componentInitParametersProtocol`
- `componentInputProtocol`
- `componentOutputProtocol`

Each field within the template schema is defined with attributes such as data type, allowed values, validation rules, and optional nested structures.

Template Field Attributes

Attribute	Type	Description
`type`	`string`	The data type of the field. Allowed values: `string`, `number`, `boolean`, `array`, `object`, `any`.
`description`	`string`	Human-readable description of the field's purpose.
`pattern`	`string`	(For strings) Regular expression the value must match.
`length`	`number`	Maximum allowed string length.
`min`, `max`	`number`	Minimum and maximum numeric values allowed.
`choices`	`array`	List of valid values (enumeration).
`max_length`	`number`	Maximum number of elements for arrays.

Attribute	Type	Description
`properties`	`object`	(For objects) Defines nested structure of key-value fields.
`items`	`object`	(For arrays) Defines schema for each element of the array.

Examples

---

componentInitSettingsProtocol

```
{
  "logging_level": {
    "type": "string",
    "description": "Logging verbosity level",
    "choices": ["debug", "info", "warn", "error"]
  },
  "enable_cache": {
    "type": "boolean",
    "description": "Whether to enable cache during execution"
  },
  "max_threads": {
    "type": "number",
    "description": "Maximum threads the component can spawn",
    "min": 1,
    "max": 64
  }
}
```

---

componentInputProtocol

```
{
  "image": {
    "type": "string",
    "description": "Base64 encoded image input",
    "pattern": "^data:image/.*;base64,"
  },
  "metadata": {
    "type": "object",
    "description": "Optional metadata associated with the input",
    "properties": {
      "source": { "type": "string" },
      "timestamp": { "type": "number" }
    }
  }
}
```

---

componentOutputProtocol

```
{
  "predictions": {
    "type": "array",
    "description": "Top prediction results",
    "max_length": 5,
    "items": {
      "type": "object",
      "properties": {
        "label": { "type": "string" },
        "confidence": { "type": "number", "min": 0.0, "max": 1.0 }
      }
    }
  },
  "runtime_ms": {
    "type": "number",
    "description": "Inference time in milliseconds"
  }
}
```

---

componentInitParametersProtocol

```
{
  "threshold": {
    "type": "number",
    "description": "Minimum confidence required to accept a prediction",
    "min": 0.0,
    "max": 1.0
  },
  "top_k": {
    "type": "number",
    "description": "Number of top predictions to return",
    "min": 1,
    "max": 10
  },
  "class_filter": {
    "type": "array",
    "description": "Limit predictions to specific class labels",
    "choices": ["cat", "dog", "car", "tree"],
    "max_length": 4
  }
}
```

Component Spec example: Object Detection

```
{
  "body": {
    "spec": {
      "values": {
        "componentId": {
          "name": "object-detector",
          "version": "2.0.0",
          "releaseTag": "stable"
        },
        "componentURI": "model.object-detector:2.0.0-stable",
        "componentType": "model",

        "containerRegistryInfo": {
          "containerImage": "registry.ai-platform.com/models/object-detector:2.0.0",
          "containerRegistryId": "ai-platform-registry",
          "containerImageMetadata": {
            "author": "vision-team",
            "description": "YOLO-based real-time object detection model"
          },
          "componentMode": "aios"
        },

        "componentMetadata": {
          "usecase": "real-time object detection",
          "framework": "PyTorch",
          "hardware": "GPU"
        },

        "componentInitData": {
          "weights_path": "/models/yolov5s.pt",
          "device": "cuda"
        },

        "componentInputProtocol": {
```

```
    "image": {
      "type": "string",
      "description": "Input image encoded as Base64",
      "pattern": "^data:image/.*;base64,"
    },
    "image_shape": {
      "type": "array",
      "description": "Height and width of the input image",
      "max_length": 2,
      "items": {
        "type": "number",
        "min": 1
      }
    }
  },

  "componentOutputProtocol": {
    "detections": {
      "type": "array",
      "description": "Detected objects",
      "items": {
        "type": "object",
        "properties": {
          "class": { "type": "string" },
          "confidence": { "type": "number", "min": 0.0, "max": 1.0 },
          "bbox": {
            "type": "array",
            "description": "Bounding box [x_min, y_min, x_max, y_max]",
            "max_length": 4,
            "items": { "type": "number" }
          }
        }
      }
    },
    "processing_time_ms": {
      "type": "number",
      "description": "Inference time in milliseconds"
    }
  },

  "componentInitParametersProtocol": {
    "threshold": {
      "type": "number",
      "description": "Confidence threshold for filtering predictions",
      "min": 0.0,
      "max": 1.0
    },
    "top_k": {
      "type": "number",
      "description": "Return top-K detections per image",
      "min": 1,
      "max": 50
    }
  },

  "componentInitSettingsProtocol": {
    "enable_tracing": {
      "type": "boolean",
      "description": "Enable performance tracing logs"
    },
    "batch_size": {
      "type": "number",
      "description": "Batch size to be used during inference",
      "min": 1,
      "max": 32
    }
  },

  "policies": {
    "resource_affinity": {
      "nodeType": "gpu",
      "minMemory": "4GB"
    }
  },

  "componentManagementCommandsTemplate": {
    "restart": {
      "description": "Restart the model service",
      "args": {}
    },
    "reload_weights": {
      "description": "Reload model weights from disk",
      "args": {
        "weights_path": {
          "type": "string",
          "required": true
        }
      }
    }
  },

  "componentParameters": {
          "threshold": 0.4,
          "top_k": 10
        },

        "componentInitSettings": {
          "enable_tracing": true,
          "batch_size": 4
        },

        "tags": ["vision", "object-detection", "yolo", "realtime"]
      }
    }
   }
  }
```

Onboarding the component:

```
curl -X POST http://<server-url>/api/addComponent \
  -H "Content-Type: application/json" \
  -d @component_spec.json
```