The IDM JSON Schema

V.1.1, 2025-06-03

Contents

The IDM JSON Schema

1 Introduction

IDM defines a standard format that manufacturers may use to provide information about their product in a structured format – thus permitting automated multi-vendor product selection, and eventually compatibility checking and microgrid design verification.

The standard format is defined in a series of JSON schema files that may be used to verify that the manufacturer-provided JSON product data is consistent with the IDM format. JSON schema files are not very descriptive or easily read, so this is a commentary on the structure, which may be used to ensure that a parameter means the same here as in any other JSON file.

The IDM Schema provides a structure for the Source Pro catalog of microgrid products. The initial objective is to support a “first-pass” filter to aid homing in on the most suitable product. The schema does not contain sufficient detail to support confirmation of electrical or functional suitability, and for this, the user must refer to the manufacturer’s provided data (which is accessible from the Source Pro product page).

Whilst many parameter values are optional, and may simply be omitted, certain parameters are considered essential, in that if missing, the data is essentially useless. These parameters are highlighted with an asterisk.

1.1 Selection tool

A selection tool may be created to facilitate the selection of suitable candidate products according to system requirements. Initially, a horizontally organized list of parameters is provided, with a list of options underneath. Multiple selections of options are permitted. As each selection is made, unsupported combinations of other parameters are greyed out. When the “Apply filters” button is clicked, the resulting selection is displayed, one row per product. The list of filter parameters is not necessarily the same as the list of displayed parameters – for example, a thumbnail image of the product is included in the display but would not be included in the filter.

It is strongly suggested that contributors to this document should familiarize themselves with the selection tools offered by Mouser, Digikey and other electronic component distributors for choosing (say) resistors or diodes. This document proposes using the same format (with different parameters, of course) for selecting commodity microgrid components.

1.2 Creation of product IDM files using AI

It is intended that creation of the JSON file describing a product using AI is supported. To achieve this, a prose question is offered, and a Regular Expression used to extract the parameter value from the returned verbose reply. *[This is not implemented in the current version.]*

2 Universal JSON Parameters

File: `common-schema.json`

The parameters which are common to all products in the catalog are defined in this file. Listed below are all the parameters that are common to all types of product. Particular types of product have their own schemas defined in separate files, which must be in the same folder, or sub-folders.

2.1 Manufacturer *

File: `manufacturer-schema.json`

Information about the manufacturer of the product, ie where to go for the most authoritative source of information about the product.

2.1.1 JSON Schema

This parameter uses the standard JSON format for describing a company or other organization (see §3.1).

2.1.2 Selection tool presentation

The company name appears in both the selection filter and the selection display, with a hyperlink to the company's main website home page.

2.2 Product Name

File: `productName-schema.json`

A name given to the product by the manufacturer.

2.2.1 JSON Schema

```
{
	"schema": {
		"type": "string",
		"minLength": 2,
		"maxLength": 254
	}
}
```

2.2.2 Selection tool presentation

The product name appears in the selection filter.

2.3 Product Identifier *

File: `productIdentifier-schema.json`

The part number assigned to the product by the manufacturer.

2.3.1 JSON Schema

```
{
        "schema": {
                "type": "string",
                "minLength": 2,
                "maxLength": 254
        }
}
```

2.3.2 Selection tool presentation

The product name appears in the selection display.

2.4 Product Series

File: `productSeries-schema.json`

The product range, family or series that this product forms part of, if any.

2.4.1 JSON Schema

```
{
        "schema": {
                "type": "string",
                "minLength": 0,
                "maxLength": 254
        },
        "selectionTool": {
                "filter": true,
                "display": false
        }
```

2.4.2 Selection tool presentation

The product series only appears in the selection filter.

2.5 Datasheet hyperlink

File: `common-schema.json`

This should be a specific hyperlink either to the manufacturer's product web page (preferred), or to a downloadable product manual.

(Note that the schema also supports the uploading of any number of files by the manufacturer.)

2.5.1 JSON Schema

```
{
        "schema": {
                "type": "string",
                "format": “uri”
        }
}
```

2.5.2 Selection tool presentation

This is used as a hyperlink behind the thumbnail image of the product in the selection tool display section.

2.6 Description

File: `common-schema.json`

This is a free text field in which a prose description of the product may be provided.

2.6.1 JSON Schema

```
{
   "description": {
      "type": "string",
      "minLength": 0,
      "maxLength": 65535
    }
}
```

2.6.2 Selection tool presentation

This is not displayed unless on a product-specific web page hyperlinked from the display row of the selection tool.

2.7 Distributors

File: `distributors-schema.json`

A list of distributors, wholesalers or importers of the (commodity) product. This may be a single company, or an array of companies, each using the company schema format defined in §3.1.

It is assumed that this will be populated by the manufacturer, to provide pointers to companies likely to hold stock.

2.7.1 JSON Schema

```
{
   "schema": {
    "anyOf": [
      {"$ref": "company-schema.json#/schema"},
      {
        "type": "array",
        "items": {"$ref": "company-schema.json#/schema"}
      }
    ]
  },
  "selectionTool": {
    "filter": false,
    "display": true
  }
}
```

2.7.2 Selection tool presentation

These may be listed (and potentially hyperlinked) from the display row of the selection tool.

2.8 Systems Integrators

File: `distributors-schema.json` (We can use the same file for this)

A list of companies that are approved systems integrators for complex products. This may be a single company, or an array of companies, each using the company schema format defined in §3.1.

2.8.1 JSON Schema

```
{
   "schema": {
    "anyOf": [
      {"$ref": "company-schema.json#/schema"},
      {
        "type": "array",
        "items": {"$ref": "company-schema.json#/schema"}
      }
    ]
  },
  "selectionTool": {
    "filter": false,
    "display": true
  }
}
```

2.8.2 Selection tool presentation

These may be listed (and potentially hyperlinked) from the display row of the selection tool.

2.9 Embargoed countries

File: `embargoedCountries-schema.json`

There may be certain end-use countries to which under US law it is not permitted to ship products. These may be listed here. *[NB The format should be agreed. It is suggested that ISO3166 country codes are used.]*

2.9.1 JSON Schema

```
{
  "schema": {
    "oneOf": [
      { "type": "string", "minLength": 2 },
      {
        "type": "array",
        "items": { "type": "string", "minLength": 2 }
      }
    ]
  }
}
```

2.9.2 Selection tool presentation

(This is not presented in either field of the selection tool.)

2.10 Not recommended for new designs

File: `common-schema.json`

If a product is End-of-Life, it should not be included in new designs, although it may still be available for replacement purposes. This is a Boolean yes/no flag.

2.10.1 JSON Schema

```
{
  "schema": {
    "type": "Boolean”
  }
}
```

2.10.2 Selection tool presentation

It is suggested that if a product is EoL, it is either not displayed at all in the display section of the selection tool, or that it is greyed out.

2.11 Type-specific parameters

File: `typeSpecifics-schema.json`

Each type of microgrid component has critical parameters that are specific to its function. These are listed separately by product type in §0.

2.12 Listing Authorities

File: `listingAuthorities-schema.json`

A list of organizations that have certified the product to meet certain standards. Each organization can create a profile in the format defined in §3.1.

2.12.1 JSON Schema

```
{
   "schema": {
    "anyOf": [
      {"$ref": "company-schema.json#/schema"},
      {
        "type": "array",
        "items": {"$ref": "company-schema.json#/schema"}
      }
    ]
  },
  "selectionTool": {
    "filter": false,
    "display": "<a href \"https:$webHomePageURL$\">$coLogo$</a>"
  }
}
```

2.12.2 Suggested initial dropdown list

This might include: UL, CSA, TUV, Nemko, CE, UKCA – though each will need an organization profile JSON file.

2.12.3 Selection tool presentation

These may be listed (and potentially hyperlinked) from the display row of the selection tool. Icons may be used in preference to organization names. Where a specific test reference is available (as for example provided by SEC in Chile), this can be hyperlinked under the icon.

2.13 Environmental parameters

File: `environmental-schema.json`

The manufacturer may declare certain environmental constraints on the safe and reliable operation of the product. These are listed separately by product type in §3.2.

2.14 Files

File: `files-schema.json`

The manufacturer may upload various product-specific files, which the microgrid designer can download.

2.14.1 Schema

```json
{
  "schema": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "filename": { "type": "string" },
        "fileType": { "type": "string" }
      }
    }
  }
}
```

2.15 Images

File: `images-schema.json`

The manufacturer may upload various pictures of the product, which the microgrid designer can view.

2.15.1 Schema

```json
{
  "schema": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "filename": {"type": "string"},
```

```
                "mimeType": { "type": "string" }
              }
            }
          }
        }
```

3 Shared Parameter Definitions

There are several parameters that appear in more than one context (for example, current limits may be rated currents, peak currents or breaking currents, but the format for defining a maximum current may be the same.

3.1 Company/organization profile data format

File: `company-schema.json`

Any organization involved in IDM can create an extensive profile for themselves, using the profile model developed for DC-IDE, including a logo. This will significantly extend this part of the schema. It should be a separate JSON file, referenced by the product-specific JSON file.

3.1.1 JSON Schema

The standard JSON format for describing a company or other organization is:

```
{
  "schema": {
    "type": "object",
    "properties": {
      "coName": {
        "type": "string",
        "minLength": 3,
        "maxLength": 254
      },
      "webHomePageURL": {
        "type": "string",
        "format": "uri"
      }
    },
    "$comment": "The logo can easily be represented here by a Base64 string",
    "coLogo": {"type": "string"},
    "required": [ "coName", "webHomePageURL"]
  },
  "selectionTool": {
    "display": "<a href \"https:$webHomePageURL$\">$coName$</a>",
    "$comment": "This will need some coding to build the hyperlink"
  }
}
```

The webHomePageURL could be the URL of the appropriate DC-IDE Profiles page. The schema for that is obviously much more extensive than this, and it probably doesn't make sense to replicate it here (even if I had it, which I don't).

It may be more valid to consider the company profile page as a marketing tool, since all relevant information about the company will already be available on the company's own website.

3.2 Mechanical Attributes

File: `mechanical-schema.json`

These comprise size, weight and mounting styles.

3.2.1 Schema

```
       "schema": {
              "type": "object",
    "properties": {
      "length": { "$ref": "dimensions-schema.json#/schema" },
      "width": { "$ref": "dimensions-schema.json#/schema" },
      "depth": { "$ref": "dimensions-schema.json#/schema" },
      "height": { "$ref": "dimensions-schema.json#/schema" },
      "diameter": { "$ref": "dimensions-schema.json#/schema" },
      "weight": {
        "type": "object",
        "properties": {
          "value": { "type": "number" },
          "unit": {
            "type": "string",
            "enum": [ "g", "kg", "oz", "lbs" ]
          }
        },
        "additionalProperties": false
      },
      "mountingType": {
        "oneOf": [
          { "$ref": "#/definitions/mountingStyleType" },
          {
            "type": "array",
            "items": { "$ref": "#/definitions/mountingStyleType" }
          },
          {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "style": { "$ref": "#/definitions/mountingStyleType" },
                "orderCode": { "type": "string" }
              }
            }
          }
        ]
      }
    }
       },

       "selectionTool": {
              "filter": false,
              "display": false
       },
       "definitions": {

              "mountingStyleType": {
                     "type": "string",
                     "enum": [
                            "floor",
                            "wall",
                            "panel",
                            "din-rail",
                            "rack"
                     ]
              }
       }
}
```

The mounting styles will certainly extend over time.

3.2.2 Physical Dimensions

File: `dimensions-schema.json`

All measurements of length can adhere to the same schema.

Dimensions may be entered as a JSON object {“Value”,”Units”) or as a string comprising numbers followed by the units, eg:

```
"diameter": {
    "value": 12.5,
    "units": "mm"
}
```

...or:

```
"diameter": "12.5mm",
```

Either format is equally valid.

3.2.2.1 JSON Schema

```
{
  "$comment": "This schema may be used for any physical length value",
  "schema": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "value": { "type": "number" },
          "units": {
            "type": "string",
            "enum": [ "in", "ft", "yds", "mi", "mm", "cm", "m", "km" ]
          }
        },
        "required": [ "value", "units" ]
      },
      {
        "type": "string",
        "pattern": "^([0-9]+(.[0-9]+)?)(in|ft|yds|mi|mm|cm|m|km)$"
      }

    ]
  }

}
```

3.2.3 Weight

3.3 Environmental data format

File: `environmental-schema.json`

The manufacturer may declare certain environmental constraints on the safe and reliable operation of the product. Specifically, this may relate to:

- Ambient operating temperature range,
- Operating Relative Humidity (as defined in IEC 60068-2-11)

- Ingress protection standards (IEC60529 and/or NEMA)
- Operating altitude
- Cooling method
- RoHS Compliance

3.3.1 JSON Schema

```json
{
  "schema": {
    "type": "object",
    "properties": {
      "operatingTemperature": {
        "type": "object",
        "properties": {
          "min": {
            "type": "number"
          },
          "max": {
            "type": "number"
          },
          "unit": {
            "type": "string",
            "enum": [ "C", "F"]
          }
        },
        "additionalProperties": false
      },
      "operatingHumidity%": {
        "type": "object",
        "properties": {
          "min": {
            "type": "number",
            "minimum": 0
          },
          "max": {
            "type": "number",
            "maximum": 100
          }
        },
        "additionalProperties": false
      },
      "ingressProtection_IP": {
        "type": "string",
        "pattern": "^IP([0-6])|x([0-9])|x[ABCD]*[HMSW]*$"
      },
      "ingressProtection_NEMA": {
        "type": "string",
        "enum": [
          "1",
          "2",
          "3",
          "3X",
          "3S",
          "3SX",
          "3R",
          "3RX",
          "4",
          "4X",
          "5",
          "6",
          "6P",
```

```
              "12",
              "12K",
              "13"
            ]
          },
          "maximumOperatingAltitude": {"$ref": "dimensions-schema.json#/schema"},

          "additionalProperties": false
          },
          "coolingMethod": {
            "type": "string",
            "enum": [
              "passive",
              "forced-air",
              "liquid",
              "none"
            ]
          },
          "RoHScompliant": {"type": "boolean"}
        }
      }
    }
```

3.4 Current rating

File: `currentRating-schema.json`

Current limits are specified at several points in the product schemas. Currents may be entered as a JSON object {“Value”,”Units”) or as a string comprising numbers followed by the units, eg:

```
"currentRating": {
    "value": 10,
    "units": "A"
}
```

...or:

```
"currentRating": "1.5kA",
```

Either format is equally valid.

3.4.1 JSON Schema

```
{
  "$comment": "This schema may be used for any component current rating",
  "schema": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "value": {"type": "number"},
          "units": {
            "type": "string",
            "enum": [ "mA", "A", "kA"]
          }
        },
        "required": [ "value", "units"]
      },
      {
        "type": "string",
```

```
            "pattern": "^([0-9]+(.[0-9]+)?)(k|m)?A$"
          }
        ]
      }
    }
```

3.5 Power rating

File: `powerRating-schema.json`

Power ratings are specified at several points in the product schemas. Powers may be entered as a JSON object {“Value”,”Units”) or as a string comprising numbers followed by the units, eg:

```
"powerRating": {
    "value": 10,
    "units": "W"
}
```

...or:

```
"powerRating": "1.5kW",
```

Either format is equally valid.

3.5.1 JSON Schema

```
"schema": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "value": { "type": "number" },
          "units": {
            "type": "string",
            "enum": [ "mW", "W", "kW" ]
          }
        },
        "required": [ "value", "units" ]
      },
      {
        "type": "string",
        "pattern": "^(-?[0-9]+(.[0-9]+)?)(k|m)?W$"
      }
    ]
```

3.6 Voltage rating

File: `voltageRating-schema.json`

Voltage limits are specified at several points in the product schemas. Voltages may be entered as a JSON object {“Value”,”Units”) or as a string comprising numbers followed by the units, eg:

```
"voltageRating": {
    "value": 600,
    "units": "mV"
}
```

...or:

```
"voltageRating": "1.25kV",
```

Either format is equally valid.

3.6.1 JSON Schema

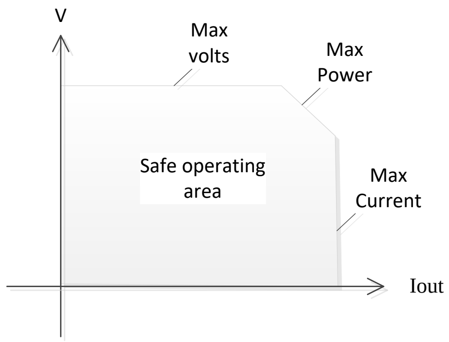
```
{
  "$comment": "This schema may be used for any component voltage rating",
  "schema": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "value": {"type": "number"},
          "units": {
            "type": "string",
            "enum": [ "mV", "V", "kV"]
          }
        },
        "required": [ "value", "units"]
      },
      {
        "type": "string",
        "pattern": "^([0-9]+(.[0-9]+)?)(k|m)?V$"
      }
    ]
  }
}
```

3.7 Port types

File: `port-schema.json`

There are several port types of importance in microgrids. They all have their own particular relationship between voltage and current and power. Typically, one of these three parameters will be controlled by hardware or software, and the other two will be determined by whatever the port is connected to.

Every port has a safe operating area, defined by limits of voltage, current and power it can handle. This may be represented in a graph:



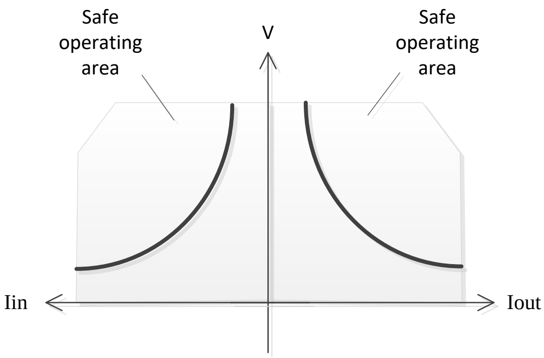
3.7.1 Schema

```
  "$comment": "This JSON schema is for any electrical power port",
  "schema": {
    "type": "object",
    "properties": {
```

```
              "portType": {
                "oneOf": [
                  { "$ref": "constPowerPort-schema.json#/schema" },
                  { "$ref": "constCurrentPort-schema.json#/schema" },
                  { "$ref": "constVoltsPort-schema.json#/schema" },
                  { "$ref": "solarInputPort-schema.json#/schema" },
                  { "$ref": "resistivePort-schema.json#/schema" },
                  { "$ref": "batteryChargingPort-schema.json#/schema" }
                ]
              },
              "portName": { "type": "string" },
              "powerDirection": {
                "type": "string",
                "enum": [ "input", "output", "bi-directional"]
              },
              "frequency": { "$ref": "frequency-schema.json#/schema" },
              "maxVoltage": { "$ref": "voltageRating-schema.json#/schema" },
              "maxCurrentOut": { "$ref": "currentRating-schema.json#/schema" },
              "maxCurrentIn": { "$ref": "currentRating-schema.json#/schema" },
              "maxPowerIn": { "$ref": "powerRating-schema.json#/schema" },
              "maxPowerOut": { "$ref": "powerRating-schema.json#/schema" },
              "connector": { "type": "string" }
            }
          },
```

3.7.2 Constant power port

File: `constPowerPort-schema.json`



In many applications, the desired power flow is determined by software, rather than (or in addition to) hardware or the electrical conditions on the ports.

Typically, a constant-power port will either be a power source or a power sink, but bidirectional ports are possible.

3.7.2.1 Voltage droop control

A particular form of constant power control is “voltage droop control”, where the voltage observed indicates the level of power required. A voltage higher than nominal indicates that the microgrid has excess power, and that the power being supplied should be reduced. Conversely, lower voltages indicate that more power should be supplied. This may be implemented in hardware or software. Particular attention must be paid to this when more than one power source is feeding a DC bus, to avoid oscillation between supplies.

3.7.2.2 *Frequency power control*

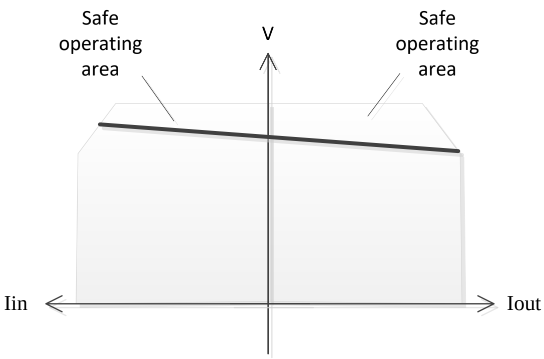
The equivalent to voltage droop control in AC microgrids is frequency control – a frequency higher than nominal indicated power over-supply, and below nominal power under-supply.

3.7.2.2.1 Schema

```
{
  "$comment": "This schema describes a port with a constant power
characteristic.  The target power level will be determined either by the need
for power balance with the other port, or by software",
  "schema": {
     "type": "object",
    "properties": {
      "portType": {
        "type": "string",
        "const": "constPower"
      },
      "maxVolts": { "$ref": "voltageRating-schema.json#/schema" },
      "maxCurrentIn": { "$ref": "currentRating-schema.json#/schema" },
      "maxCurrentOut: { "$ref": "currentRating-schema.json#/schema" },
      "powerLimitOut": { "$ref": "powerRating-schema.json#/schema" },
      "powerLimitIn": { "$ref": "powerRating-schema.json#/schema" }
    }
  }
}
```

3.7.3 Constant voltage port

File: `constVoltsPort-schema.json`



A constant-voltage port will deliver (or draw in) whatever current will maintain the voltage at the level set by the hardware of the component. (The controlled value may be DC, or single- or polyphase AC, with a constant RMS voltage.) There will be a maximum current determined by the power limitations of the converter, beyond which constant voltage control breaks down.

In practice, there will always be a small variation in terminal voltage as the current varies. The general strategy is to try to minimize this.

$$V_{out} = (V_{oc} - IR_{thevenin})$$

Typically, a constant-voltage port will either be a power source or a power sink, but bidirectional ports are possible.

Exceptionally, the target constant voltage may be determined by manual adjustment or as in the case of USB, by digital communication and software.

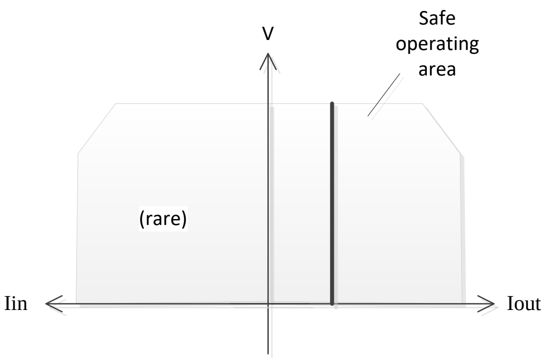
3.7.3.1.1 Schema

```
{
  "$comment": "This schema describes a port with a constant voltage characteristic.",
  "schema": {
     "type": "object",
    "properties": {
      "portType": {
        "type": "string",
        "const": "constantVolts"
      },
      "openCctVolts": { "$ref": "voltageRating-schema.json#/schema" },
      "maxCurrentOut": { "$ref": "currentRating-schema.json#/schema" },
      "maxCurrentIn": { "$ref": "currentRating-schema.json#/schema" },
      "powerLimitOut": { "$ref": "powerRating-schema.json#/schema" },
      "powerLimitIn": { "$ref": "powerRating-schema.json#/schema" },
      "theveninResistance": { "$ref": "resistanceRating-schema.json#/schema" }
    }
  }
}
```

3.7.4 Specialized ports

3.7.4.1 Constant Current

File: `constCurrentPort-schema.json`



A constant-current port will either source or sink the current at a level specified by the hardware. If sourcing current, the voltage will be set to deliver the required current. There will be a maximum voltage the supply will support in trying to maintain the specified current (for example into an open circuit).

Constant-current supplies are important for delivering uniform brightness in LED lighting. Constant current may also be important for some battery charging (see below). Constant current loads are rare in microgrids.

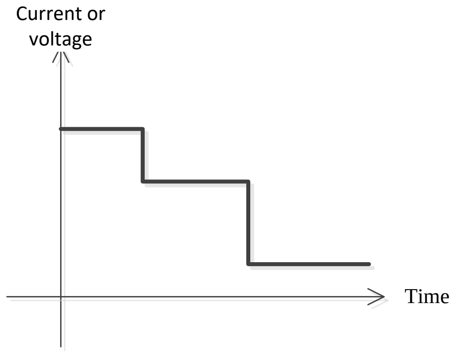
3.7.4.1.1 Schema

```
{
  "$comment": "This schema describes a port with a constant current characteristic.",
  "schema": {
```

```
          "type": "object",
         "properties": {
           "portType": {
             "type": "string",
             "const": "constantCurrent"
           },
           "targetCurrentOut": { "$ref": "currentRating-schema.json#/schema" },
           "openCctVolts": { "$ref": "voltageRating-schema.json#/schema" }
         }
       }
     }
```

3.7.4.2 Battery charging output

File: `batteryChargingPort-schema.json`



(Note the horizontal axis here is 'time'.)

A battery charger output port will implement a charging regime defined by the battery chemistry or the battery manufacturer, typically based on values and curves of the battery voltage, but perhaps by a Battery Management System (BMS). Certain voltages and currents will be imposed for particular durations. Some of the time, a constant-voltage characteristic may be presented instead of constant-current. The values may also be a function of battery temperature.

The rate of charging may be determined by the limitations of available power.

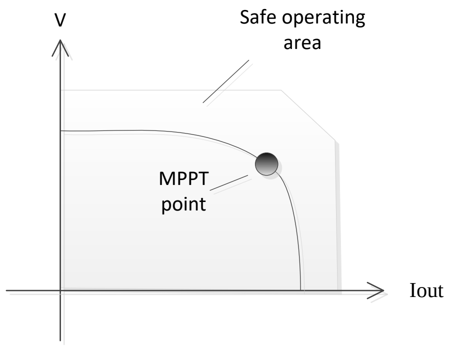
3.7.4.2.1 Schema

```
{
  "schema": {
    "$comment": "This schema is for a battery charging port",
    "type": "object",
    "properties": {
      "portType": {
        "type": "string",
        "const": "constantVolts"
      },
      "nominalVoltage": { "$ref": "voltageRating-schema.json#/schema" },
      "chemistry": { "type": "string" },
      "BMScommunication": {
        "type": "object",
        "properties": {
          "protocol": {
            "anyOf": [
              { "type": "string" },
              {
                "type": "array",
```

```
                    "items": { "type": "string" }
                  }
                ]
              }
            },
            "interface": {
              "anyOf": [
                { "$ref": "string" },
                {
                  "type": "array",
                  "items": { "type": "string" }
                }
              ]
            }
          }
        },
        "required": [ "nominalVoltage" ]
      }
    }
```

3.7.4.3 Solar input port

File: `solarInputPort-schema.json`



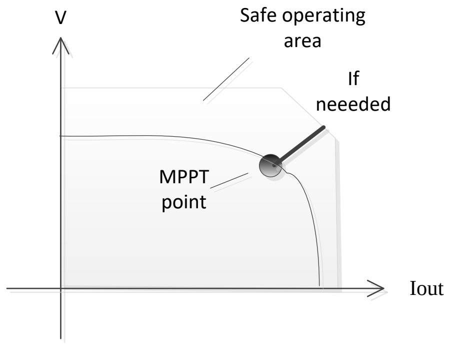
A solar input port will adjust the input conditions in order to maximize the power extracted from the solar panel(s), typically using an MPPT algorithm.

3.7.4.3.1 Schema

```
{
  "$comment": "This schema describes a solar or wind input port for connection
to solar panels.  In the case of wind/hydro, there is no requirement for the
input power to equal the output power.",
  "schema": {
     "type": "object",
    "properties": {
      "portType": {
        "type": "string",
        "const": "solarInput"
      },
      "maxOpenCctVolts": { "$ref": "voltageRating-schema.json#/schema" },
      "maxShortCctCurrent": { "$ref": "currentRating-schema.json#/schema" },
      "maxPower": { "$ref": "powerRating-schema.json#/schema" },
    }
  }
}
```

3.7.4.4 Wind/hydro input port

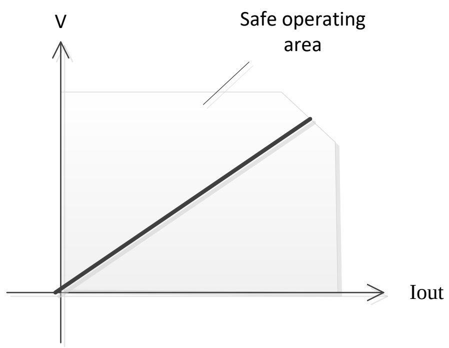
File: `solarInputPort-schema.json` (The solar panel schema may be used.)



A wind/hydro input port will also attempt to maximize the power extracted, but if the power exceeds the permissible level, energy must be dissipated in a local load resistor, since disconnection would result in the rotor spinning out of control.

3.7.4.5 Resistive Load

File: `resistivePort-schema.json`



Both the oldest and probably the least important port type from a microgrid point of view!

3.7.4.5.1 Schema

```
{
  "$comment": "This schema describes a port with a resistive characteristic",
  "schema": {
     "type": "object",
    "properties": {
      "portType": {
        "type": "string",
        "const": "resistive"
      },
      "resistance": { "$ref": "resistanceRating-schema.json#/schema" },
      "maxPower": { "$ref": "powerRating-schema.json#/schema" }
    }
  }
```

3.8 Mounting Style

File: `mountingStyle-schema.json`

Most microgrid components expect to be fixed to something.

3.8.1 Initial Suggested dropdown list of mounting options

- "Surface mount",
- "DIN rail",
- "Panel mount",
- "Wall-mount",
- "Free-standing"

3.9 Cables

File: `wireSizes-schema.json`

Cable sizes have to be specified in several contexts in microgrid design.

3.9.1 Initial dropdown values

The standard sizes are:

- "30AWG, 0.05mm2",
- "28AWG, 0.08mm2",
- "26AWG, 0.14mm2",
- "24AWG, 0.25mm2",
- "22AWG, 0.34mm2",
- "21AWG, 0.38mm2",
- "20AWG, 0.50mm2",
- "18AWG, 0.75mm2",
- "17AWG, 1.0mm2",
- "16AWG, 1.5mm2",
- "14AWG, 2.5mm2",
- "12AWG, 4.0mm2",
- "10AWG, 6.0mm2",
- "8AWG, 10mm2",
- "6AWG, 16mm2",
- "4AWG, 25mm2",
- "2AWG, 35mm2",
- "1AWG, 50mm2",
- "1/0AWG, 55mm2",
- "2/0AWG, 70mm2",
- "3/0AWG, 95mm2"

If necessary, other sizes can be added, and differentiation between solid, seven- or thirteen-stranded, and fine-stranded cable.

3.10 Bolt sizes

File: `boltSize-schema.json`

There are many different standards for bolt threads. For microgrid purposes, the only important parameter is the required hole size for the eyelet terminating the cable.

3.10.1 Initial dropdown values

The standard sizes are:

```
M6 (1/4"),
M8 (5/16"),
M10 (13/32"),
M12 (1/2")
```

3.11 Connection Style

File: `connection-schema.json`

Most microgrid components are permanently wired in place via clamping screw terminals (onto wire ends or ferrules) or by eyelets onto bolts. Screw terminals are specified by the wire capacity they can accommodate, bolts by the outside diameter of the bolt, which will require the eyelet to be slightly larger.

3.11.1 Connection schema

```json
{
  "$comment": "Options for connecting microgrid components.",
  "schema": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "bolt": { "$ref": "boltSize-schema.json#/schema" }
        },
        "required": [ "bolt" ]
      },
      {
        "type": "object",
        "properties": {
          "terminal": { "$ref": "wireSizes-schema.json#/schema" }
        },
        "required": [ "terminal" ]
      }
    ],

  }

}
```

Wire sizes are defined in §3.9. Bolt sizes are defined in §3.10.

4 Type-specific parameters

Each type of microgrid component has critical parameters that are specific to its function.

4.1 Fuses

File: `fuse-schema.json`

In this case, the assumption is that a fuse is a replaceable component, and therefore must be accessible. The physical shape is important, but obviously the current rating is the critical factor from an electrical point of view.

4.1.1 Fuse shape and physical size *

File: `fuseType-schema.json`

Fuses come in many shapes and sizes. For microgrids, the most popular formats are “Cartridge”, “Flush square body”, “Blade”, “L25S/L50S” and “SQB”, and the dropdown list will initially be populated with these – but inevitably, others will need adding. (We do not consider fuses that are soldered in place.) Automotive blade fuses are popular for low-power 12V and 24Vdc microgrids. Each format has a range of sizes – but each format uses its own terminology.

4.1.2 Current Rating *

File: `currentRating-schema.json`

The maximum continuous load current the fuse will pass indefinitely without blowing.

The format is defined in §3.4.

4.1.3 Maximum Breaking Current

File: `currentRating-schema.json`

The maximum fault current the fuse will interrupt.

The format is defined in §3.4.

4.1.4 Maximum Breaking Voltage (AC/DC)

File: `voltageRating-schema.json`

The maximum voltage across the fuse terminals after the fuse has blown. As there is a real possibility of an arc between the ends of the broken fuse element, this voltage is always equal or less for DC than for AC.

The format is defined in §3.5.

4.1.5 Fuse speed of response

File: `fuseResponse-schema.json`

Fuses allow a certain amount of energy to pass in excess of the rated current before they blow. Slow-blow fuses tolerate more than fast-blow fuses, which in turn tolerate more than fuses designed to protect semiconductors.

4.1.5.1 Initial schema values

```
"Semiconductor",
"Fast blow (F)",
"Normal (M)",
"Slow blow (T)",
"Time delay (TT)"
```

Manufacturers may add to this list. For further information, see IEC60269, or https://www.swe-check.com.au/pages/learn_fuse_speed.php. A more quantitative treatment would use the I^2T characteristic curves supplied by the manufacturer.

4.1.6 Blown fuse indicator

Some fuses are provided with an indicator that changes color or appearance when the fuse blows. This is just a Boolean parameter indicating whether such functionality is present.

4.1.6.1 JSON Schema

```
{
  "schema": {
    "type": "Boolean”
  }
}
```

4.2 Breakers

File: `breaker-schema.json`

A breaker opens a circuit if excessive current flows. It may also function as a manual on/off switch. After an overcurrent has occurred, most breakers require a manual reset, but a few are “reclosers”, closing again automatically two or three times in case the fault has cleared itself, or are resettable remotely.

4.2.1 Trip Criteria *

Breakers are designed to interrupt the current if an anomalous situation occurs. This may be:

- Overcurrent
- Under – or Over-Voltage
- Leakage to ground (“ground fault”)
- Arcing
- Phase imbalance
- Manual turn-off by a user

Some breakers can also be tripped by an external solenoid.

The schema allows for each of these trip mechanisms to be specified in any combination. However, breakers are often referred to by acronyms according to their trim mechanism(s) – though note that this usage is not always consistent, and the terminology in Europe differs from that in the US:

	GFCI	AFCI	RCB	RCD	RCBO	RCCB	MCB	MCCB	AFCB	ELCB
Overcurrent					Y		Y	Y		
Over/undervoltage										
Ground leakage	Y		Y	Y	Y	Y				Y

Arcing		Y							Y	

Therefore, the schema permits one of the above acronyms in lieu of specifying the criteria individually.

4.2.1.1 JSON Schema

```
{
  "$comment": "Reasons that a breaker might turn off.",
  "schema": {
    "type": "object",
    "parameters": {
      "overCurrent": { "type": "boolean" },
      "overVoltage": { "type": "boolean" },
      "underVoltage": { "type": "boolean" },
      "groundFault": { "type": "boolean" },
      "arcFault": { "type": "boolean" },
      "manualOperation": { "type": "boolean" },
      "phaseImbalance": { "type": "boolean" },
      "externalSolenoid": { "type": "boolean" },
      "acronym": {
        "type": "string",
        "enum": [
          "GFCI",
          "AFCI",
          "RCB",
          "RCD",
          "RCBO",
          "RCCB",
          "MCB",
          "MCCB",
          "AFCB",
          "ELCB"
        ]
      }
    }
  }
}
```

4.2.2 Detection technology

Several detection methods are in common use, each with particular strengths and weaknesses.

The two principal methods for detecting overcurrent are magnetic (the current in a coil attracts an armature) or thermal (a bimetallic strip heats up) and in either case, the resulting movement releases a catch. The coil acts fast, the bimetallic strip is slower. Where the permissible surge current duration is several seconds, an additional hydraulic damper may slow the process further.

Some breakers employ electronics to detect anomalous conditions (this is always true for arc fault detection, and over/undervoltage detection).

4.2.2.1 Dropdown list of detection methods

- "Thermal",
- "Thermal-magnetic",
- "Magnetic",
- "Thermal-magnetic-hydraulic",

- "Electronic",
- "Hybrid"

This list is probably complete until some new technology is developed.

4.2.3 Number of Poles *

A breaker can interrupt a number of current-carrying conductors simultaneously (for example, a three-phase breaker may interrupt the three live conductors, or those and the neutral).

4.2.3.1 Schema

```
{
  "schema": {
    "type": "object",
    "properties": {
      "numberOfPoles": {
        "type": "integer",
        "minimum": 1
      },
```

4.2.4 Current Rating *

The maximum steady-state current the breaker will allow. As tripping follows a curve gradually reducing the time taken to trip as the current over the rated current increases, a current marginally over the rated current *could* trip the breaker, but it might take a very long time.

The format for the current is defined in §3.4.

4.2.5 Voltage Rating AC/DC

When the breaker is closed, the voltage across the terminals is minimal, but when the breaker contacts open, the full supply voltage is presented across them. If the contacts are mechanical, there will be some arcing, which will be short-lived if the supply is AC, but could continue indefinitely with DC. Therefore the voltage rating for DC will always be lower than for AC. Some manufacturers specify an increased DC voltage by connecting two opening poles in series to double the arc length.

4.2.5.1 Voltage Rating Schema

```
"voltageRatingAC": { "$ref": "voltageRating-schema.json#/schema" },
"voltageRatingDC": { "$ref": "voltageRating-schema.json#/schema" },
"voltageRatingDC-2PolesInSeries": {"$ref": "voltageRating-schema.json#/schema"},
```

The format for the voltage is defined in §3.5.

4.2.6 Breaking Capacity

When a short-circuit occurs, the current may initially be very large – many times the maximum current the breaker is designed to allow to pass. This parameter specifies the maximum breaking current the breaker can interrupt.

The format for the current is defined in §3.4.

4.2.7 Overcurrent Trip Curves

There are many loads that require a high current briefly when first powered up – for example, large motors. Breakers are chosen according to the degree of overcurrent and its duration that are required not to trip the breaker.

4.2.7.1 Initial Suggested dropdown list of trip curves

- `"IEC 60947-2 Type Z",`
- `"IEC 60898-1 Type B",`
- `"IEC 60898-1 Type C",`
- `"IEC 60947-2 Type K",`
- `"IEC 60898-1 Type D",`
- `"IEC 60947-2 Type MA",`
- `"IEC 60934",`
- `"Custom"`

4.2.8 Isolation Mechanism

Breakers can interrupt the current either by opening mechanical contacts, or by turning off solid-state semiconductors, or by a combination of the two.

4.2.8.1 JSON Schema

```
"isolationMechanism": {
  "type": "string",
  "enum": [ "mechanical", "solid-state", "hybrid" ]
},
```

This is probably a complete list until some new technology arrives.

4.2.9 Mounting style

This is defined in §3.7.

4.2.10 Connections

This is defined in §3.10.

4.2.11 Reset mechanism

After a breaker has tripped, it must be reset to restore the connection. There are really only three options:

- Manual (the default if not specified)
- Auto – this mostly applies to reclosers for high voltage systems, which will try to restore the current two or three times before giving up
- Remote – an external command sent via some communications method

4.2.11.1 Schema

```
"reset": {
  "type": "string",
  "enum": [ "manual", "auto", "remote" ]
```

4.2.12 Auxiliary contact

Some breakers have an auxiliary contact, which may be used to trigger an alarm, or ensure that some other equipment does not remain powered after the breaker has tripped. Normally-closed (NC), Normally-open (NO) and Changeover (C/O) auxiliary contacts are possible.

4.2.12.1 Schema

```
                "auxiliaryContact": {
                  "type": "string",
                  "enum": [ "NC", "NO", "C/O" ]
```

4.3 Solar Panels

Solar panels are the most popular renewable energy source for DC microgrids.

4.3.1 Panel Type *

File: `panelType-schema.json`

4.3.1.1 Schema

```
        "schema": {
              "type": "string"
          },
          "initialValues": [ "Building-integrated monofacial", "Flexible",
        "Bifacial","Monofacial","Mono/bi-facial" ]
```

Building-integrated panels are panels designed to replace roof tiles/slates, rather than to be mounted above them (referred to as "building-applied").

Monofacial panels only generate power from the sun shining on one side. Bifacial panels also generate some power from the back of the panel. This may give an uplift to the power output of 5-30%. Mono/bi-facial panels are intended to be used either with or without any sun shining on the back. Flexible panels can conform to modestly-curved surfaces.

There will doubtless emerge other types of solar panel, which will need to be added to this list in due course.

4.3.2 Panel Technology

File: `panelTech-schema.json`

The microgrid designer may not be concerned with how the panel has been manufactured, as long as it does the job required.

4.3.2.1 Schema

```
          "schema": {
              "type": "string"
          },
          "initialValues": [ "Monocrystalline", "Polycrystalline", "Thin-film",
        "Perovskite" ]
        }
```

There are several material types used to manufacture solar panels. These are the important ones.

4.3.3 Electrical Characteristics

Since the electrical characteristics of a solar panel vary with temperature, they are typically quoted either as "STC" (Standard Temperature Conditions) or "NMOT" (Normal Module Operating Temperature) or "NOCT" (Normal Operating Cell Temperature). Although both the latter refer to an irradiance of 800 W/m^2, an ambient air temperature of 20°C, and a wind speed of 1 m/s, they are defined slightly differently, but each aspires to provide a more realistic performance in practice than STC (Standard Test Conditions, 25°C, irradiance of 1000 W/m^2).

For a “first-pass” selection process to select a better (or cheaper but equally good) product, it makes sense to compare like-for-like, for example to compare STC values for one with STC values for the other – even if neither product will actually meet these values in practice.

Regardless of the test conditions used, the performance figures quoted are:

4.3.3.1 Watts Peak *

The maximum power the panel is capable of generating (schema in §3.5). This is perhaps the most important parameter for a solar panel.

4.3.3.2 Open Circuit Volts *

The maximum voltage the panel can generate under no load (schema in §3.6). This is important for specifying the solar charge controller, as it will potentially have to withstand this voltage.

4.3.3.3 Short Circuit Current

The maximum current the panel can generate, when fed into a short circuit (schema in §3.43.6).

4.3.3.4 MPPT Volts and Current

The output voltage and current at the Maximum Power Point Tracking load conditions (when the peak wattage is being produced). This gives a more realistic value for the typical operating conditions when exposed to plenty of sunlight.

4.3.3.5 Efficiency

The percentage (0...100) of the incident radiation power that is converted to electricity. A figure of 20% is typical.

4.3.4 Bifacial Gain 5...30%

For bifacial panels, the electrical performance is enhanced by incident solar radiation on the back of the panel. This will normally be a fraction of the radiation hitting the front (a perfect mirror reflecting 100% of the sunlight hitting it to the back of the panel would give 100% bifacial gain, and double the power output).

4.3.5 Maximum System Voltage

Typically, several solar panels will be connected in series, raising the voltage to ground. The quality of the insulation around the panels determines the maximum voltage to ground that will be considered safe. The schema is defined in §3.63.6).

4.3.6 Maximum Fuse Rating

There will be a limit to the current the solar cells and cell interconnects can safely carry, regardless of any other factors. The string of panels should be fused by a fuse with a current rating no greater than this (schema in §3.43.6).

4.3.7 Integral Bypass Diode

Bypass diodes, also known as free-wheeling diodes, are wired within the PV module and provide an alternate current path when a cell or panel becomes shaded or faulty. They may or may not be included.

4.3.7.1 Schema

```
"integralBypassDiode": { "type": "boolean" },
```

4.3.8 Performance Warranty Years

Some manufacturers guarantee that their products will not degrade to more than a certain percentage within a certain number of years.

4.3.8.1 Schema

```
"performanceWarranty": {
  "years": { "type": "number" },
  "percentageReducedTo": {
    "type": "number",
    "minimum": 0,
    "maximum": 100
  }
}
```

4.3.9 Mechanical Attributes *

Clearly, the size and shape of a solar panel is of critical importance – the number of panels is usually determined by the available area to mount them. However, the dimensions and weight can be defined in the same way as any other product. Therefore, this can use the schema defined in §3.23.6).

4.3.10 Environmental parameters for solar panels

The environmental operating conditions (temperature, humidity, etc) that apply to any other product also apply to solar panels, and the definition in §3.3 may be used.

Environmental parameters specific to solar panels include the weight of snow per square metre they are guaranteed to survive, and the incident wind speed. These are not accounted for in this schema, as they will be unusual parameters to base product selection on.

4.3.11 Connector

All electrical components of a microgrid will have connections to other components via some kind of terminal or connector. The schema already caters for bolt terminations and screw clamp terminals, but solar panels typically are provided with MC3 or MC4-compatible single-pole connectors, with the female connector on the positive solar panel terminal (the polarity must be reversed for the connection to a solar charge controller). In this schema, a simple string defines the supplied connectors.

4.3.11.1 Initial dropdown list

Either "MC3" or "MC4". Manufacturers must be empowered to add further connector types.

4.4 Batteries

Batteries, flow batteries and fuel cells are used to provide energy storage for microgrids. As they represent a very significant fraction of the cost and space requirement of a microgrid installation, they are an area of very active technological development, and any schema must be ready to accept new technologies as they are made available. In this section, we focus solely on batteries, with a single bidirectional pair of terminals presenting the DC battery voltage (ie excluding products that include inverters, separate charging ports etc, but including batteries that have an integral battery management system (BMS) to ensure that all cells of the battery contribute equally.

4.4.1 Nominal Voltage *

Batteries always have a quoted nominal voltage, which is usually somewhere near the middle of the typical voltage range of the battery. This is an essential first parameter when selecting a suitable product. The standard voltage rating definition in §3.6 may be used.

Clearly, for establishing electrical compatibility, the full possible voltage range will be important.

4.4.2 Energy capacity *

The amount of energy the battery can store is also an important parameter. In practice, this is a function of:

- Battery temperature
- How fast the battery is charged and discharged
- How deep a discharge the user is willing to make the battery endure (almost all battery technologies suffer if the battery is discharged completely

In comprehensive battery documentation, curves will be supplied detailing how these parameters affect the energy stored.

In addition, the amount of energy put into the battery will be greater than the amount given out – the rest being dissipated as heat (or less desirably as permanent degradation of the internal chemistry). This is the so-called “round-trip efficiency”.

Despite all these caveats, it is essential that the manufacturer should provide an indication of the amount of energy the user should expect to get. The convention is that discharge capacity is quoted, at a certain discharge rate (eg C10 means discharging at a rate that discharges from full to empty in ten hours). A fast discharge (eg C1) will produce a much lower total energy capacity than a very slow discharge (eg C100). If a discharge rate is not quoted, C10 may be assumed.

The energy may be quoted in amp-hours (Ah), which can be multiplied by the nominal battery voltage to give the energy stored (watt-hours, Wh or kWh).

4.4.2.1 Schema

```
{
  "$comment": "This schema may be used for battery energy capacity",
  "schema": {
    "oneOf": [
      {
        "type": "object",
        "properties": {
          "value": { "type": "number" },
          "units": {
            "type": "string",
            "enum": [ "Ah", "Wh", "kWh" ]
          }
        },
        "required": [ "value", "units" ]
      },
      {
        "type": "string",
        "pattern": "^([0-9]+(.[0-9]+)?)(Ah|Wh|kWh)$"
      }
    ],
    "dischargeRate": {
```

```
            "type": "string",
            "pattern": "^C([0-9]+(.[0-9]+)?$"
          }
        }
      }
```

4.4.3 Chemistry *

There is a whole taxonomy of battery chemistries, and new ones are appearing almost daily. As each has its own strengths and weaknesses, selecting the optimal technology for a particular application becomes very important. Critical factors include:

- Safety issues (fire, outgassing, electrolyte spill, toxicity etc)
- Energy density (kWh/kg) – this will determine size and weight for a given energy capacity
- Cost (of course)
- Guaranteed number of discharge cycles (to a given discharge depth)
- Operating temperature range, and the impact of temperature on energy capacity

Once a preferred battery chemistry has been selected, it is essential that the electronics to charge the battery are configured to prevent overcharging, typically by setting the charging regime for the particular chemistry. It is also important for the control electronics to limit the discharge to the desired minimum charge level selected to optimize battery life against usable storage capacity.

For the purposes of this schema, the battery chemistry is simply a string, with a suggested dropdown list of initial values. The electronics associated with the battery should be selected to support the same technology (perhaps by means of manual configuration of voltages and charge times).

4.4.3.1 Suggested initial battery chemistry choices

Information taken from https://batteryuniversity.com/ .

- Lead-acid
  - Flooded (Wet)
  - VRLA (Valve-regulated Lead-Acid)
    - Standard, sealed
    - AGM (Absorbent Glass Mat)
    - Gel, carbon-gel
- Lithium (graphite anode)
  - Lithium Iron Phosphate ($LiFePO_4$)
  - Lithium Cobalt Oxide (LCO)
  - Lithium Manganese Oxide (LMO)
  - Lithium nickel manganese cobalt oxide (NMC)
  - Lithium nickel cobalt aluminum oxide (NCA)
- Lithium Titanate anode
  - Lithium nickel manganese cobalt oxide (NMC)
  - Lithium nickel cobalt aluminum oxide (NCA)
- Nickel Metal Hydride (NiMH)
- Nickel-Cadmium
- Sodium-ion

4.4.4 Battery terminals

Many companies have used the same definitions for battery terminals, viz:

Auto Post Terminal (SAE terminal)

This is the most common battery terminal type, and any person who has replaced a car battery can easily recognize it. In order to prevent accidently connecting the terminals in reverse polarity, the positive post is always larger diameter than the negative. Another terminal that you will find is what is known as Pencil Post (found predominantly in batteries for Japanese cars – JIS types). When compared with a SAE terminal, the Pencil Post is smaller.

Stud Terminal

This is a 3/8" threaded stainless steel terminal is designed to fasten and hold the terminal connection to the terminal lug onto the lead base of the terminal.

Dual Post Terminal / Marine Terminal

This terminal type has an Automotive Post and a Stud (5/16"). You can make the connection using either a traditional pressure contact or a ring terminal and wing nut connection.

Button Terminal

These are also known as insert terminals. You will find these terminals from M5 to M8 which refers to the metric size of the diameter of the bolt thread. For example, if you have a battery with a M8 terminal, you will need a bolt with an 8 millimetre diameter thread. These types of terminals are most commonly found on Absorbed Glass Mat batteries used in emergency backup and uninterruptable power systems (UPS) battery applications.

AT Terminal (Dual SAE / Stud type terminals)

They are commonly found in traction type batteries used in heavy cycling applications such as floor scrubbers and off-grid solar application batteries. This terminal type has an Automotive Post and a Stud (3/8" threaded stainless steel terminal).

I have therefore added an “other” option to the connection-schema (see §3.11), with the only suggested dropdown value as “SAEterminal”. (Stud and Button can both use the “bolt” value.)

4.4.5 Battery Management System

Certain battery chemistries have the characteristic that cells connected in series may not balance automatically (certain cells taking more charge than others), leading to some cells being overcharged while others are undercharged. Battery management systems (BMS) exist to correct this, and to control the overall amount and rate of charge. This is essential to prevent batteries overheating, with potentially disastrous consequences.

A BMS will communicate with the battery charger (and potentially discharger) to ensure that charging is managed correctly. The communications interface and protocol need to be specified.

4.4.5.1 Schema

```
"BMScommunication": {
  "type": "object",
  "properties": {
    "protocol": {
      "anyOf": [
        { "type": "string" },
```

```
                {
                  "type": "array",
                  "items": { "type": "string" }
                }
              ]
            }
          },
          "interface": {
            "anyOf": [
              { "$ref": "string" },
              {
                "type": "array",
                "items": { "type": "string" }
              }
            ]
          }
        }
```

Note that even if the interface and protocol data checks out, there will remain many fine details of the communication protocol that could give rise to incompatibility between the charge/discharge controller and the battery.

4.5 Converters and Inverters

A converter or inverter[1] is a two-port component, without integral energy storage, and where one electrical characteristic (voltage, current or power) of one of the ports is controlled by hardware or software. It connects part of the system over which it has no control to part of the system that needs a parameter controlled.

With no energy storage capability, the hardware must ensure that the power input follows the power output with a certain efficiency, plus some static losses to power the internal hardware, the difference $(P_{in} - P_{out})$ being lost as heat:

$$P_{in} = \frac{P_{out}}{Efficiency} + P_{static}$$

At least one of the ports must have a constant-power characteristic, in order for the hardware to be able to balance the above equation. The converter will typically set whatever current is necessary to ensure that the required power flows.

If the converter is unable to balance the above equation, it must shut down, or compromise its port control regime so that it can.

The key parameters for a converter are therefore:

- What controls the amount of power that is converted – this may be defined by what the converter is connected to (the load or power source on the controlled port), or by external control or software.

[1] An inverter is a converter in which one port is DC and the other AC, and power flow from DC port to AC port is supported.

- The voltage/current/power characteristics of the controlled port (the other will be constant-power)
- The safe voltage/current/power limits of the two ports,
- Whether each port is DC or AC, and if AC, how the frequency is determined.

A control port may also be present, which determines the behavior of the converter. This will have a physical interface, and a communications protocol.

4.5.1 Schema

```
    "$comment": "This JSON schema is for 2-port power converters and inverters,
  but no energy storage",
    "schema": {
      "type": "object",
      "properties": {
        "port1": {
          "$ref": "port-schema.json#/schema"
        },
        "port2": {
          "$ref": "port-schema.json#/schema"
        },
        "staticPower": {
          "$ref": "powerRating-schema.json#/schema"
        },
        "transferPowerSetBy": {"type": "string", "enum":
  ["port1","port2","firmware","controlPort","the lower of P1 and P2"]}
        "controPort": { "$ref": "controlPort-schema.json#/schema" },
        "transferEfficicency%P1toP2": {
          "type": "number",
          "minimum": 0,
          "maximum": 100
        },
        "transferEfficicency%P2toP1": {
          "type": "number",
          "minimum": 0,
          "maximum": 100
        }
      }
```

4.5.2 Common microgrid converters

	Power flow set by	**Controlled Port**	**Bi-directional?**
AC-DC Power supply	DC load	Contant voltage DC (output)	No
DC-AC Inverter	AC load	Constant voltage AC (output)	No
Solar charge controller	MPPT algorithm at low solar power, output voltage at high power	MPPT solar input (low power), Output port (high power)	No
Solar inverter	MPPT algorithm at low solar power, frequency at high power	MPPT solar input (low power), Output port (high power)	No
AC or DC Battery charger	Battery algorithm	Battery charger	No
Grid-tie inverter	Software	Constant power (both ports)	Maybe
LED Driver	Hardware	Constant current DC (output)	No

Solar battery charger	MPPT algorithm at low solar power, battery algorithm at high power	MPPT solar input at low solar power, battery charger at high power	No

4.6 Electric Vehicle Charging Points

An EV charging point (Electric Vehicle Supply Equipment, EVSE) is a special example of a two-port converter. Most commonly, EVSEs are unidirectional, charging the vehicle battery from the supply (in our case, a DC microgrid), but bi-directional products are slowly emerging that can use the vehicle battery to provide local storage for a small microgrid (this is usually referred to as "Vehicle-to-Home", V2H), and depending on the situation, it may also be used to provide local storage to support for the microgrid or grid as a whole (V2G). In all cases, internal firmware or software determines the behavior of the product.

4.6.1 EVSE Power levels

The power level of an EVSE can vary from 2.4kW up to 130kW or more, and this will be reflected in the time taken to charge the vehicle fully. These are referred to as:

- Level 1 (L1) – power sourced from a 120Vac socket. This will be power-limited by the circuit breaker, eg 20A (= 2.4kW)
- Level 2 (L2) – power sourced from a domestic 240V socket, or hard-wired into a domestic installation
- Leve 3 (L3) – DC fast charging

Naturally, an EV battery is DC, and in order to be able to charge your EV at home, the vehicle includes an AC-powered battery charger. However, this is power-limited to reduce weight and cost. Faster charging can be achieved by feeding DC directly to the battery, in which case the power is limited by the charging point and the amount of power it has access to. This is generally restricted to public charging points, and most of these are powered from the AC grid. However, there are obvious efficiencies to be achieved by using DC power if it is available, and of course IDM focuses on these.

4.6.2 EVSE Signaling protocols

Standard protocols are emerging for communication between the charger and the vehicle – these include:

- Open Charging Point Protocol – several versions are in use (OCPP, IEC 63584)[2]
- IEC 63110

These EV – EVSE protocols are distinct from the protocols used for charging users for the use of public charging points, and protocols for managing the total demand from a parking lot with several charging points.

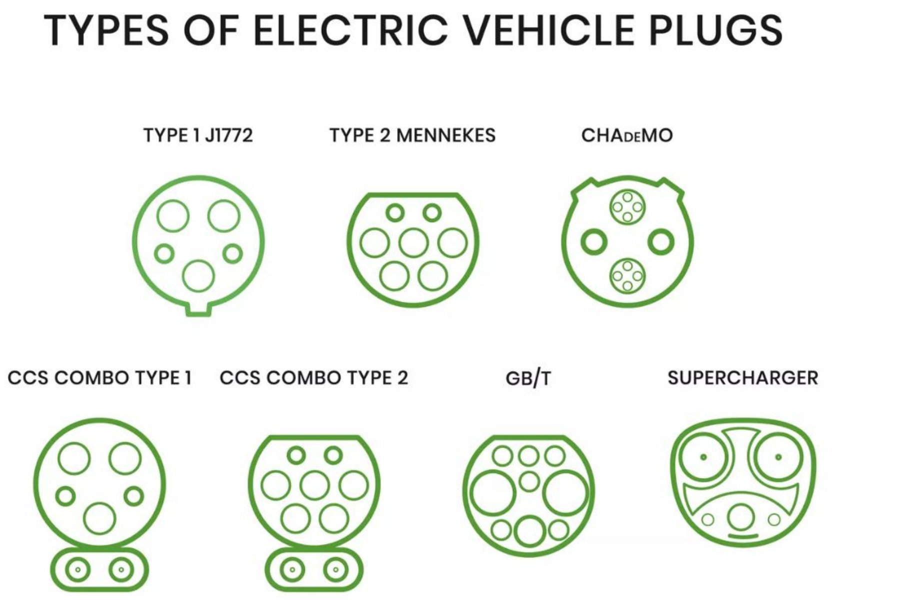
4.6.3 EV Charging Connectors

Specialized connectors are used for EV charging[3]. These include:

[2] See https://webstore.iec.ch/en/publication/95734
[3] See https://www.lifewire.com/every-ev-charging-standard-and-connector-type-explained-5201160

	L1 (AC)	L2 (AC)	L3 (DC)
J1772	√	√	
Mennekes	√	√	
Noth American Charging Standard, NACS (Tesla Supercharger)	√	√	√
Combined Charging System (CCS1 and CCS2) (backwards-compatible with J1772/Mennekes)	√	√	√
GB/T (China)	√	√	√
CHAdeMO / JEVS (usage declining)	√	√	√

A picture of these connectors may be helpful:



[4]

4.6.4 EVSE Control Ports

A typical EVSE will have one or more control port(s) supporting digital communication to allow real-time control of charging. The physical interface must be defined, and the communications protocol to be employed. The schema for a control port has been enhanced to reflect this.

4.6.5 Including EV Charging in IDM

In order to include EVSEs in the IDM schema, the definition of a port must be extended to include the particular connectors and communication protocols employed in this situation. However, these are simply additional dropdown options within the existing schema.

The one exception is that EV chargers can deal with a wide range of nominal battery voltages. Two additional parameters are therefore added to the port definition: minNominalBattVolts and maxNominalBattVolts.

It is noted that DC-IDE puts the vehicle port information under the heading “Electrical – General”, whereas it would more clearly belong in a separate output port definition.

[4] Permission for reproduction of this image has not been sought.

4.6.5.1 IDM EVSE Example


```json
{
  "$schema": "common-schema.json#/schema",
  "productName": "60kW DC-DC Mobile EV Charger",
  "productIdentifier": "AMP-8002-60",
  "description": "This data is taken from https://dcide.app/products/60kw-dc-
dc-mobile-ev-charger-amp-8002-60-dc-dc-mobile-ev-charger-
676d68cdc8a1498c3e181318#electrical",
  "manufacturer": {
    "coName": "AmperneXt",
    "webHomePageURL": "https://www.ampernext.com"
  },
  "datasheetHyperlink": "https://www.ampernext.com/products/60kw-mobile-dc-dc-
ev-charger-dc-input/",
  "notRecommendedForNewDesigns": false,
  "typeSpecificParameters": {
    "componentType": "converter",
    "port1": {
      "portType": "constantPower",
      "portName": "input",
      "powerDirection": "input",
      "frequency": "DC",
      "nominalVolts": "650V",
      "minVolts": "300V",
      "maxVolts": "820V",
      "maxCurrentIn": "100A",
      "powerLimitIn": "60kW"

    },
    "port2": {
      "portType": "batteryCharging",
      "portName": "output",
      "powerDirection": "output",
      "minNominalBattVolts": "150V",
      "maxNominalBattVolts": "1000V",
      "frequency": "DC",

      "maxCurrentOut": "200A",
      "maxPowerOut": "60kW",
      "connector": [
        {
          "type": "CCS1",
          "gender": "socket"
        },
        {
          "type": "CCS2",
          "gender": "socket"
        }
      ]
    },
    "controlPort": {
      "interface": [
        "10/100 Mbps Ethernet",
        "Wi-Fi",
        "3G/LTE"
      ],
      "controlProtocol": [ "OCPP 1.6j", "Modbus TCP" ]
    },
    "staticPower": "300W",
    "transferPowerSetBy": "controlPort",
    "transferEfficicency%P1toP2": 95
  },
```

```json
    "environmental": {
      "ingressProtection_IP": "IP54",
      "DomesticComponentRequirement": false,
      "operatingTemperature": {
        "min": -25,
        "max": 55,
        "unit": "C"
      },
      "operatingHumidity%": {
        "max": 95
      }
    },

    "mechanical": {
      "dimensions": {
        "length": 870,
        "height": 480,
        "width": 670,
        "unit": "mm"
      },

      "weight": {
        "value": 100,
        "unit": "kg"
      }
    }

  }
```