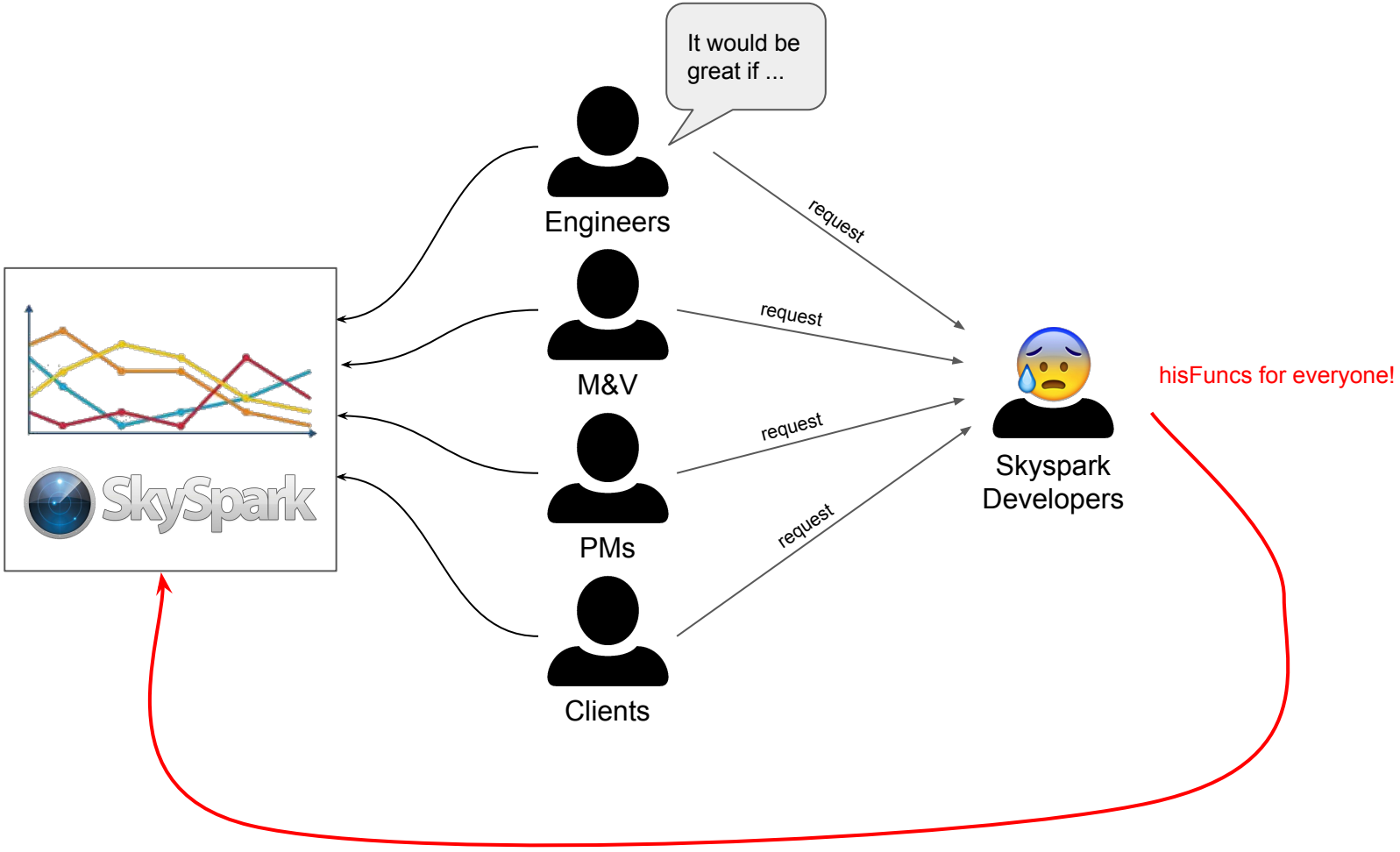
Calculation Engine

Richard Seaman
Crowley Carbon


It would be great if ...
Engineers
M&V
PMs
Clients
request
request
request
request
Skyspark Developers
hisFuncs for everyone!
SkySpark

Engineers M&V

PMs Clients

fx =B3+C3

	A	B	C	D
1	ts	Energy 1	Energy 2	Total
2		*kWh*	*kWh*	*kWh*
3	2019-10-11 00:00:00	10	15	=B3+C3
4	2019-10-11 00:05:00	13	14	27
5	2019-10-11 00:10:00	15	8	23
6	2019-10-11 00:15:00	10	14	24
7	2019-10-11 00:20:00	1	11	12
8	2019-10-11 00:25:00	3	11	14

Skyspark Developers

Func

```
(rec, dates, opts, yield) => do

  energyPt1: readById(@p:proj:r:1)
  energyPt2: readById(@p:proj:r:2)

  pts: [energyPt1, energyPt2]

  pts.hisRead(dates, opts).each(row => do

    total: row->v0 + row->v1
    yield(row->ts, total)

  end)

end
```

```
calc: {
    formula: “energy1 + energy2”
}
```

```
inputs: [
    {
        name: “energy1”,
        pointRef: @p:proj:r:1
    },
    {
        name: “energy2”,
        pointRef: @p:proj:r:2
    },
]
```

[@p:proj:r:1, @p:proj:r:2].hisRead ...

ts	energy1	energy2
2019-10-11 00:00:00	10kWh	15kWh
2019-10-11 00:05:00	13kWh	14kWh
2019-10-11 00:10:00	15kWh	8kWh
2019-10-11 00:15:00	10kWh	14kWh
2019-10-11 00:20:00	1kWh	11kWh
2019-10-11 00:25:00	3kWh	11kWh

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	10kWh + 15kWh	25kWh
2019-10-11 00:05:00	13kWh + 14kWh	27kWh
2019-10-11 00:10:00	15kWh + 8kWh	23kWh
2019-10-11 00:15:00	10kWh + 14kWh	24kWh
2019-10-11 00:20:00	1kWh + 11kWh	12kWh
2019-10-11 00:25:00	3kWh + 11kWh	14kWh

```
calc: {
    formula: “energy1 + energy2”
}
```

```
inputs: [
    {
        name: “energy1”,
        pointRef: @p:proj:r:1,
        toUnit: “kWh”
    },
    {
        name: “energy2”,
        pointRef: @p:proj:r:2,
        toUnit: “kWh”
    },
]
```

[@p:proj:r:1, @p:proj:r:2].hisRead ...

ts	energy1	energy2
2019-10-11 00:00:00	10kWh	15000Wh
2019-10-11 00:05:00	13kWh	14000Wh
2019-10-11 00:10:00	15kWh	8000Wh
2019-10-11 00:15:00	10kWh	14000Wh
2019-10-11 00:20:00	1kWh	11000Wh
2019-10-11 00:25:00	3kWh	11000Wh

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	10kWh.to(1kWh) + 15000Wh.to(1kWh)	25kWh
2019-10-11 00:05:00	13kWh.to(1kWh) + 14000Wh.to(1kWh)	27kWh
2019-10-11 00:10:00	15kWh.to(1kWh) + 8000Wh.to(1kWh)	23kWh
2019-10-11 00:15:00	10kWh.to(1kWh) + 14000Wh.to(1kWh)	24kWh
2019-10-11 00:20:00	1kWh.to(1kWh) + 11000Wh.to(1kWh)	12kWh
2019-10-11 00:25:00	3kWh.to(1kWh) + 11000Wh.to(1kWh)	14kWh

```
calc: {
    formula: “energy1 + energy2”
}
```

```
inputs: [
    {
        name: “energy1”,
        pointRef: @p:proj:r:1,
        toUnit: “kWh”,
        nullValue: 0kWh
    },
    ...
]
```

[@p:proj:r:1, @p:proj:r:2].hisRead ...

ts	energy1	energy2
2019-10-11 00:00:00	10kWh	15000Wh
2019-10-11 00:05:00		14000Wh
2019-10-11 00:10:00	15kWh	8000Wh
2019-10-11 00:15:00	10kWh	14000Wh
2019-10-11 00:20:00		11000Wh
2019-10-11 00:25:00	3kWh	11000Wh

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	10kWh.to(1kWh) + 15000Wh.to(1kWh)	25kWh
2019-10-11 00:05:00	0kWh.to(1kWh) + 14000Wh.to(1kWh)	14kWh
2019-10-11 00:10:00	15kWh.to(1kWh) + 8000Wh.to(1kWh)	23kWh
2019-10-11 00:15:00	10kWh.to(1kWh) + 14000Wh.to(1kWh)	24kWh
2019-10-11 00:20:00	0kWh.to(1kWh) + 11000Wh.to(1kWh)	11kWh
2019-10-11 00:25:00	3kWh.to(1kWh) + 11000Wh.to(1kWh)	14kWh

```
calc: {
    formula: “energy1 + energy2”
}
```

```
inputs: [
    {
        name: “energy1”,
        pointRef: @p:proj:r:1,
        toUnit: “kWh”,
        nullValue: 0kWh,
        foldFunc: “sum”
    },
    ...
]
```

[@p:proj:r:1, @p:proj:r:2].hisRead …
hisRollup ...

ts	energy1	energy2
2019-10-11 00:00:00	10kWh	15000Wh
2019-10-11 01:00:00		14000Wh
2019-10-11 02:00:00	15kWh	8000Wh
2019-10-11 03:00:00	10kWh	14000Wh
2019-10-11 04:00:00		11000Wh
2019-10-11 05:00:00	3kWh	11000Wh

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	10kWh.to(1kWh) + 15000Wh.to(1kWh)	25kWh
2019-10-11 01:00:00	0kWh.to(1kWh) + 14000Wh.to(1kWh)	14kWh
2019-10-11 02:00:00	15kWh.to(1kWh) + 8000Wh.to(1kWh)	23kWh
2019-10-11 03:00:00	10kWh.to(1kWh) + 14000Wh.to(1kWh)	24kWh
2019-10-11 04:00:00	0kWh.to(1kWh) + 11000Wh.to(1kWh)	11kWh
2019-10-11 05:00:00	3kWh.to(1kWh) + 11000Wh.to(1kWh)	14kWh

```
calc: {
   formula: “power1”
}
```

```
inputs: [
   {
      name: “power1”,
      pointRef: @p:proj:r:1,
      toUnit: “kW”,
      nullValue: 0kW,
      foldFunc: “avg”,
      convertFunc: “powerToEnergy”
   }
]
```

powerToEnergy:

```
(val, interval) => do

  (val.to(1kW)*(interval.to(1hr)))
    .as(1kWh)

end
```

[@p:proj:r:1].hisRead … hisRollup ...

ts	power1
2019-10-11 00:00:00	1000W
2019-10-11 01:00:00	
2019-10-11 02:00:00	800W
2019-10-11 03:00:00	1500W
2019-10-11 04:00:00	2100W
2019-10-11 05:00:00	700W

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	1000W.to(1kW).powerToEnergy(1hr)	1kWh
2019-10-11 01:00:00	0W.to(1kW).powerToEnergy(1hr)	0kWh
2019-10-11 02:00:00	800W.to(1kW).powerToEnergy(1hr)	0.8kWh
2019-10-11 03:00:00	1500W.to(1kW).powerToEnergy(1hr)	1.5kWh
2019-10-11 04:00:00	2100W.to(1kW).powerToEnergy(1hr)	2.1kWh
2019-10-11 05:00:00	700W.to(1kW).powerToEnergy(1hr)	0.7kWh

```
calc: {
    formula: “power1”,
    toUnit: “Wh”
}
```

```
inputs: [
    {
        name: “power1”,
        pointRef: @p:proj:r:1,
        toUnit: “kW”,
        nullValue: 0kW,
        foldFunc: “avg”,
        convertFunc: “powerToEnergy”
    }
]
```

powerToEnergy:

```
(val, interval) => do

  (val.to(1kW)*(interval.to(1hr)))
    .as(1kWh)

end
```

[@p:proj:r:1].hisRead … hisRollup ...

ts	power1
2019-10-11 00:00:00	1000W
2019-10-11 01:00:00	
2019-10-11 02:00:00	800W
2019-10-11 03:00:00	1500W
2019-10-11 04:00:00	2100W
2019-10-11 05:00:00	700W

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	(1000W.to(1kW).powerToEnergy(1hr)).to(1Wh)	1000Wh
2019-10-11 01:00:00	(0W.to(1kW).powerToEnergy(1hr)).to(1Wh)	0Wh
2019-10-11 02:00:00	(800W.to(1kW).powerToEnergy(1hr)).to(1Wh)	800Wh
2019-10-11 03:00:00	(1500W.to(1kW).powerToEnergy(1hr)).to(1Wh)	1500Wh
2019-10-11 04:00:00	(2100W.to(1kW).powerToEnergy(1hr)).to(1Wh)	2100Wh
2019-10-11 05:00:00	(700W.to(1kW).powerToEnergy(1hr)).to(1Wh)	700Wh

```
calc: {
    formula: “1.5 * wbt + 0.3 * prod”,
    asUnit: “kWh”
}
```

```
inputs: [
    {
        name: “wbt”,
        pointRef: @p:proj:r:1,
        toUnit: “°C”,
        foldFunc: “avg”
    },
    ...
]
```

[@p:proj:r:1].hisRead …
hisRollup ...

ts	wbt	prod
2019-10-11 00:00:00	15°C	50L
2019-10-11 01:00:00	16°C	45L
2019-10-11 02:00:00	17°C	55L
2019-10-11 03:00:00	16°C	52L
2019-10-11 04:00:00	17°C	48L
2019-10-11 05:00:00	16°C	49L

.map ...

eval(formula)

ts	formula	calcVal
2019-10-11 00:00:00	(1.5 * 15°C.to(1°C).as(1) + 0.3 * 50L.to(1L).as(1)).as(1kWh)	37.5kWh
2019-10-11 01:00:00	(1.5 * 16°C.to(1°C).as(1) + 0.3 * 45L.to(1L).as(1)).as(1kWh)	37.5kWh
2019-10-11 02:00:00	(1.5 * 17°C.to(1°C).as(1) + 0.3 * 55L.to(1L).as(1)).as(1kWh)	42kWh
2019-10-11 03:00:00	(1.5 * 16°C.to(1°C).as(1) + 0.3 * 52L.to(1L).as(1)).as(1kWh)	39.6kWh
2019-10-11 04:00:00	(1.5 * 17°C.to(1°C).as(1) + 0.3 * 48L.to(1L).as(1)).as(1kWh)	39.9kWh
2019-10-11 05:00:00	(1.5 * 16°C.to(1°C).as(1) + 0.3 * 49L.to(1L).as(1)).as(1kWh)	38.7kWh

calc: {
formula: String,
asUnit: String,
toUnit: String,
convertFunc: (val, interval) => newVal
}

pointInput: {
name: String,
pointRef: Ref,
toUnit: String,
nullValue: Number,
foldFunc: FoldFunc,
convertFunc: (val, interval) => newVal
}

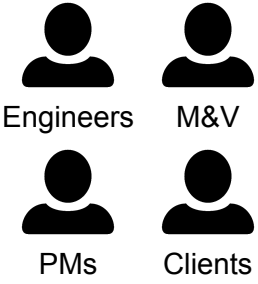
calcInput: {
name: String,
calcRef: Ref,
toUnit: String,
nullValue: Number,
foldFunc: FoldFunc,
convertFunc: (val, interval) => newVal
}

constantInput: {
name: String,
constantVal: Number,
constantInterval: Interval
}

fx powerToEnergy

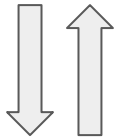
	C	D	E	F	G	H	J	K	L
1	**formula**	**asUnit**	**convertFunc**	**toUnit**	**inputName**	**point**	**convertFunc**	**toUnit**	**nullVal**
2	energy1 + energy2	kWh		Wh					
3					energy1	Site 1 - Elec Meter A - Energy			0
4					energy2	Site 1 - Elec Meter B - Power	powerToEnergy		0
5									
6									
7									

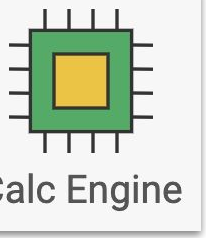
powerToEnergy
flowToTotal



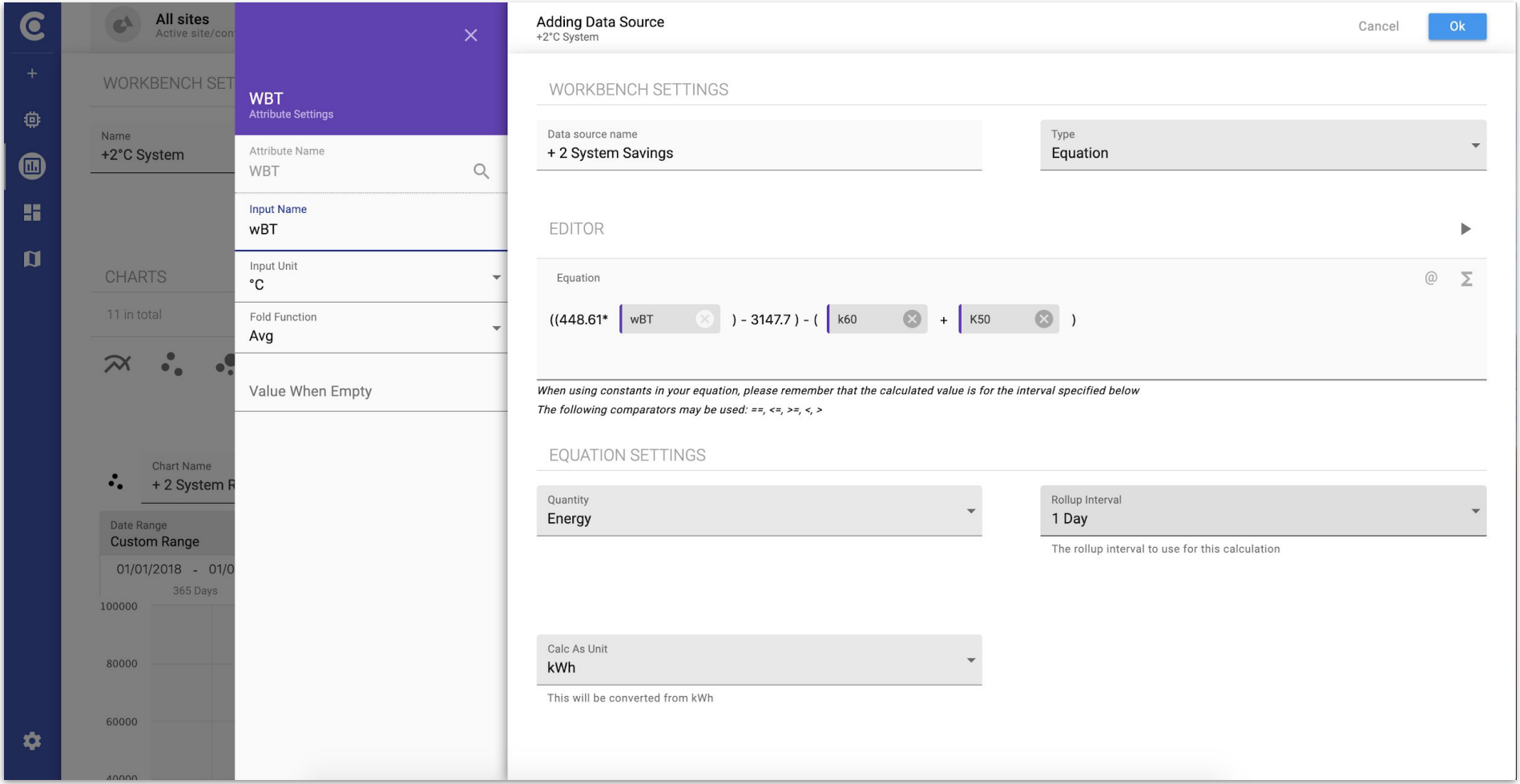
Skyspark Developers

	C	D	E	F	G	H	J	K	L
1	**formula**	**asUnit**	**convertFunc**	**toUnit**	**inputName**	**point**	**convertFunc**	**toUnit**	**nullVal**
2	energy1 + energy2	kWh							
3					energy1	Site 1 - Elec Meter A - Energy			0
4					energy2	Site 1 - Elec Meter B - Power	powerToEnergy		0





- link sheet to site
- populate sheet with available points
- validate calc / input definitions
- CRUD operations
- debugging (view calc results alongside inputs)


All sites
WORKBENCH SET
Name
+2°C System
CHARTS
11 in total
Chart Name
+ 2 System R
Date Range
Custom Range
01/01/2018 - 01/0
365 Days
100000
80000
60000
WBT
Attribute Settings
Attribute Name
WBT
Input Name
wBT
Input Unit
°C
Fold Function
Avg
Value When Empty
Adding Data Source
+2°C System
Cancel
Ok
WORKBENCH SETTINGS
Data source name
+ 2 System Savings
Type
Equation
EDITOR
Equation
((448.61* wBT) - 3147.7) - (k60 + K50)
When using constants in your equation, please remember that the calculated value is for the interval specified below
The following comparators may be used: ==, <=, >=, <, >
EQUATION SETTINGS
Quantity
Energy
Rollup Interval
1 Day
The rollup interval to use for this calculation
Calc As Unit
kWh
This will be converted from kWh