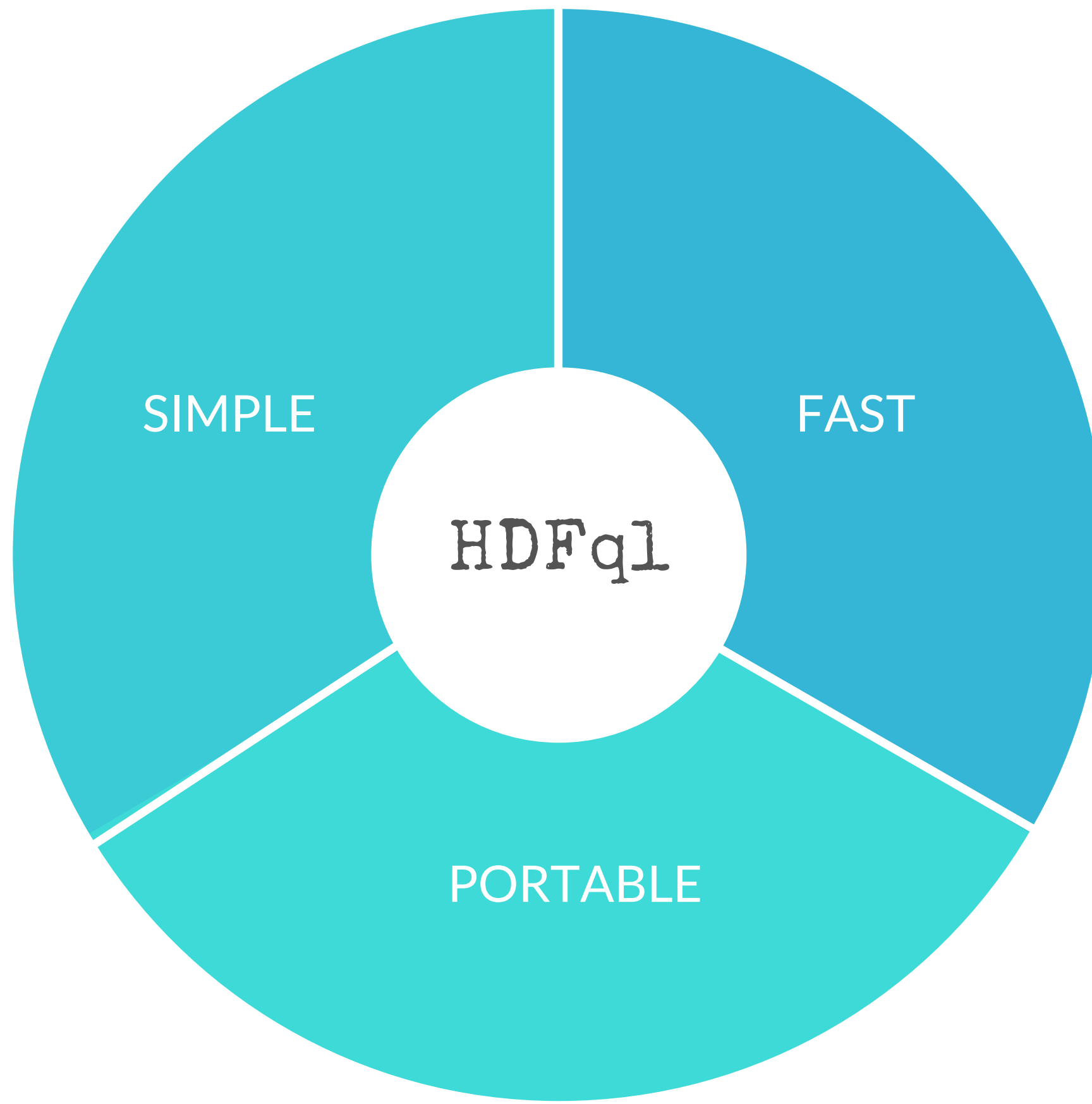




# HDFql

the easy way to manage HDF5 data



# WHAT



SIMPLE

The first tool to manage HDF5 with a language as simple and powerful as SQL



FAST

Greatly reduces the learning effort and time needed to handle HDF5



PORTABLE

Reads and writes HDF5 across programming languages and platforms



FREE

100% free (built by developers who care)



# WHO

Launched by a team of interdisciplinary experts and engineers with vast experience from world leading research facilities such as CERN, industry (Fortune 50), and leading universities in the U.S. & Europe

Strengthened by continuous feedback from the HDF5 user community to deliver solutions of high usability



**Advance  
data-driven  
discoveries**

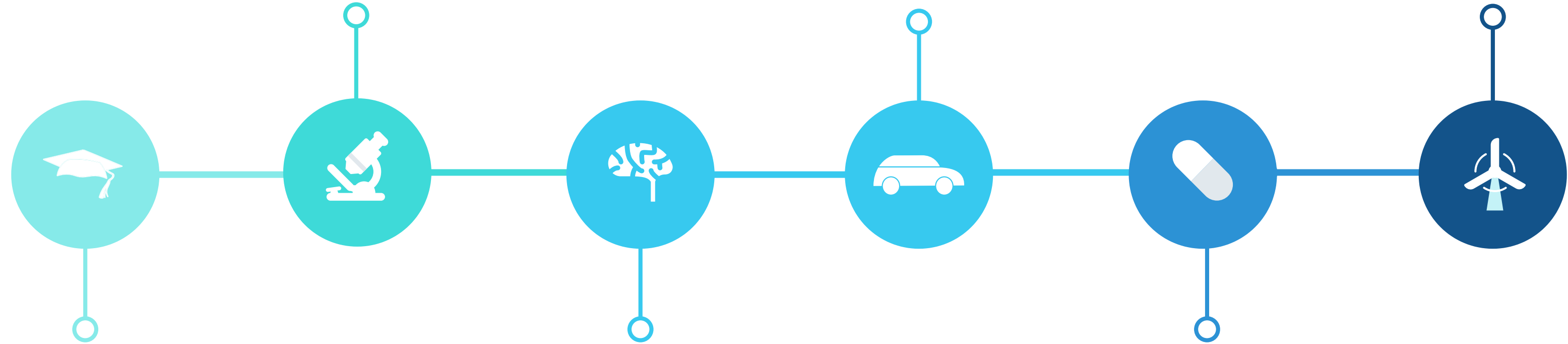
Because we seek to advance  
data-driven discoveries,  
HDFqI is free

# USERS

Research Centers

Electric Vehicle Industry

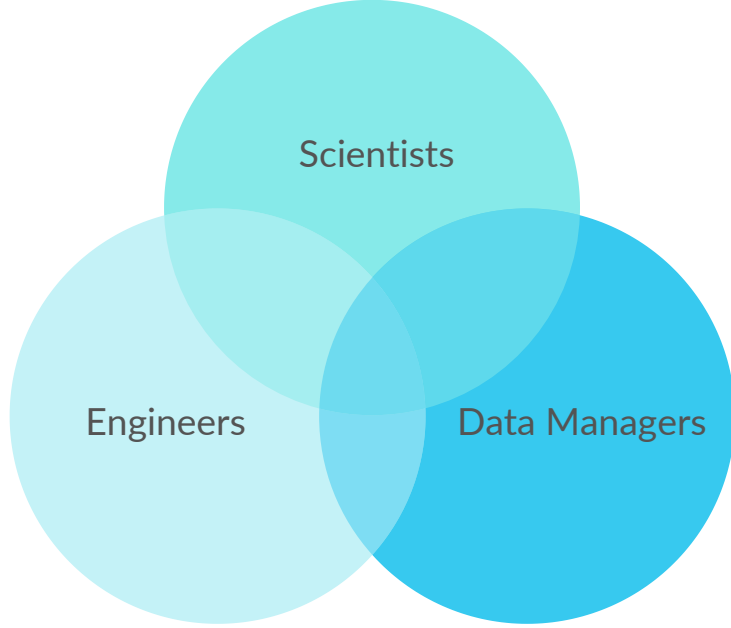
Renewable Energy



Universities

Neuroscience Consortia

Biotech Industry



# HOW

## HDFqI

```
create file myFile.h5  
use file myFile.h5  
create dataset myGroup/myDataset as int[3] enable zlib values[4, 8, 6]
```

## C API

```
hid_t file;  
hid_t group;  
hid_t dataspace;  
hid_t property;  
hid_t dataset;  
hsize_t dimension;  
int value[3];  
file = H5Fcreate("myFile.h5", H5F_ACC_EXCL, H5P_DEFAULT, H5P_DEFAULT);  
group = H5Gcreate(file, "myGroup", H5P_DEFAULT, H5P_DEFAULT, H5P_DEFAULT);  
dimension = 3;  
dataspace = H5Screate_simple(1, &dimension, NULL);  
property = H5Pcreate(H5P_DATASET_CREATE);  
H5Pset_chunk(property, 1, &dimension);  
H5Pset_deflate(property, 9);  
dataset = H5Dcreate(group, "myDataset", H5T_NATIVE_INT, dataspace, H5P_DEFAULT, property, H5P_DEFAULT);  
value[0] = 4;  
value[1] = 8;  
value[2] = 6;  
H5Dwrite(dataset, H5T_NATIVE_INT, H5S_ALL, H5S_ALL, H5P_DEFAULT, &value);
```



# HOW

## DATA DEFINITION LANGUAGE

ACTION

HDFql

Create a file named "my\_file.h5"



```
CREATE FILE my_file.h5
```

Create a file named "experiment.h5" in parallel (i.e. using MPI)



```
CREATE PARALLEL FILE experiment.h5
```

Create a group named "countries"



```
CREATE GROUP countries
```

Create a compressed dataset named "values" of data type float of two dimensions (size 250x50)



```
CREATE DATASET values AS FLOAT(250, 50) ENABLE ZLIB
```



# HOW

## DATA INTROSPECTION LANGUAGE

ACTION

HDFql

Show (i.e. get) all objects existing in current group



```
SHOW
```

Show (i.e. get) all datasets existing in group "my\_group"



```
SHOW DATASET my_group/
```

Show (i.e. get) all objects recursively starting from current group



```
SHOW LIKE **
```

Show (i.e. get) all attributes recursively starting from group "group2" that contain "1" or "3" in their names



```
SHOW ATTRIBUTE group2 LIKE **/1|3
```





# HOW

## DATA MANIPULATION LANGUAGE

ACTION

HDFql

Select (i.e. read) data from dataset "values" and populate cursor in use with it



```
SELECT FROM values
```

Insert (i.e. write) values 2.5, 3.8 and 7.9 into dataset "my\_dataset"



```
INSERT INTO my_dataset VALUES(2.5, 3.8, 7.9)
```

Insert (i.e. write) values from file "values.txt" into dataset "measurements"



```
INSERT INTO measurements VALUES FROM FILE values.txt
```

Select (i.e. read) 3rd value of dataset "dset" (using a point selection) and write its content into a user-defined variable (previously registered and assigned to number 0)



```
SELECT FROM dset(3) INTO MEMORY 0
```



# CURRENT FEATURES



Reads and writes HDF5 data



Supports parallel HDF5 (i.e. HDF5 + MPI)



Post-processes data using all cores available (using OpenMP)



Works in all major platforms (Windows, Linux, macOS)



Portable across languages (C, C++, Java, Python, C#, Fortran, R)



# FUTURE FEATURES

Support compound datasets/attributes

Support virtual datasets (VDS)

Support single-write multiple-readers (SWMR)

Support dynamically loaded filters

Pre/post-process data using BLAS library on-the-fly



# SUMMARY



## Simple

Designed to be as simple as SQL. Hides complex operations and frees users from low-level details



## Clean

Offers a clean interface requiring just a few intuitive statements even for complex operations. Gone are the days where HDF5 required endless lines of code



## Fast

Processes data using all nodes and cores available. This means much higher volumes of data are processed in the same amount of time



## Intuitive

Based on models of human cognition and natural language. Fast learning curve



## Reliable

Unfailing robustness and reliability. HDFqI is checked against hundreds of existing tests before new versions are released



## Writes

Unlike other tools, HDFqI not only reads HDF5 but also allows you to write HDF5 data



## Portable

Portable across C, C++, Java, Python, C#, Fortran and R using one uniform high-level language



## Platforms

Supports Windows, Linux, macOS



# HDFql

the easy way to manage HDF5 data

New users, partnerships or potential sponsors always welcome

[www.hdfql.com](http://www.hdfql.com)

