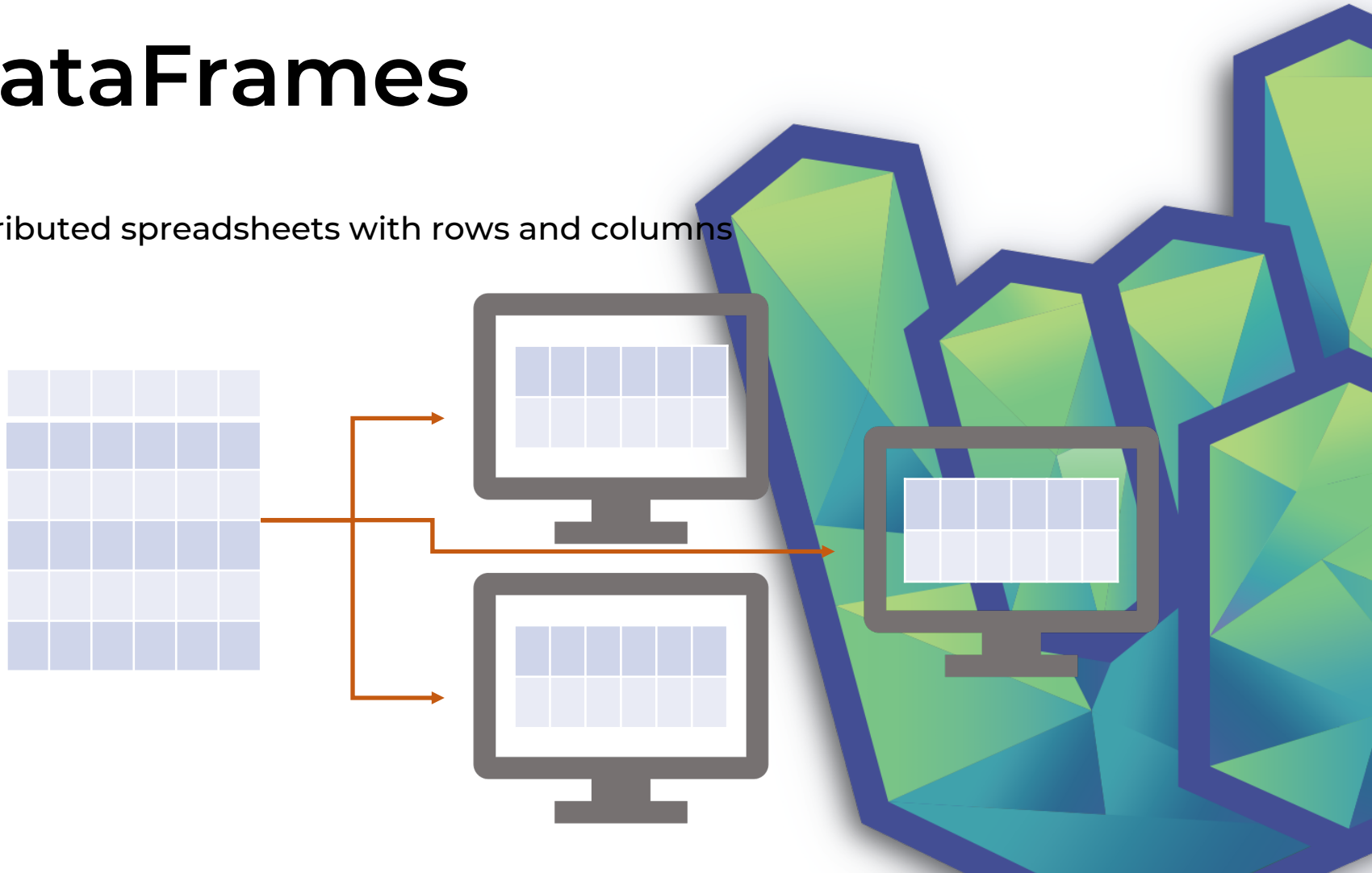
DataFrames

DataFrames

Distributed spreadsheets with rows and columns



DataFrames

Distributed collections of Rows conforming to a schema

Schema = list describing the column names and types

- types known to Spark, not at compile time
- arbitrary number of columns
- all rows have the same structure

```
val carsSchema = StructType(Array(
  StructField("Name", StringType),
  StructField("HorsePower", IntegerType),
  StructField("Acceleration", DoubleType)
))
```

Need to be distributed

- data too big for a single computer
- too long to process the entire data on a single CPU

Partitioning

- splits the data into files, distributed between nodes in the cluster
- impacts the processing parallelism

DataFrames

Immutable

- can't be changed once created
- create other DFs via transformations

Transformations

- narrow = one input partition contributes to at most one output partititon (e.g. map)
- wide = input partitions (one or more) create many output partitions (e.g. sort)

Shuffle = data exchange between cluster nodes

- occurs in wide transformations
- massive perf topic

Computing DataFrames

Lazy evaluation

- Spark waits until the last moment to execute the DF transformations

Planning

- Spark compiles the DF transformations into a graph before running any code
- logical plan = DF dependency graph + narrow/wide transformations sequence
- physical plan = optimized sequence of steps for nodes in the cluster
- optimizations

Transformations vs Actions

- transformations describe how new DFs are obtained
- actions actually start executing Spark code

Spark rocks