MVC IN ACTION: CHARTS

MODEL
STORES THE DATA UNDERLYING
THE USER INTERFACE

UPDATES

UPDATES

VIEW
HAS THE VISUAL
REPRESENTATION
OF THE DATA

UPDATES

CONTROLLER
INTERACTS WITH THE USER
AND MODIFIES THE MODEL
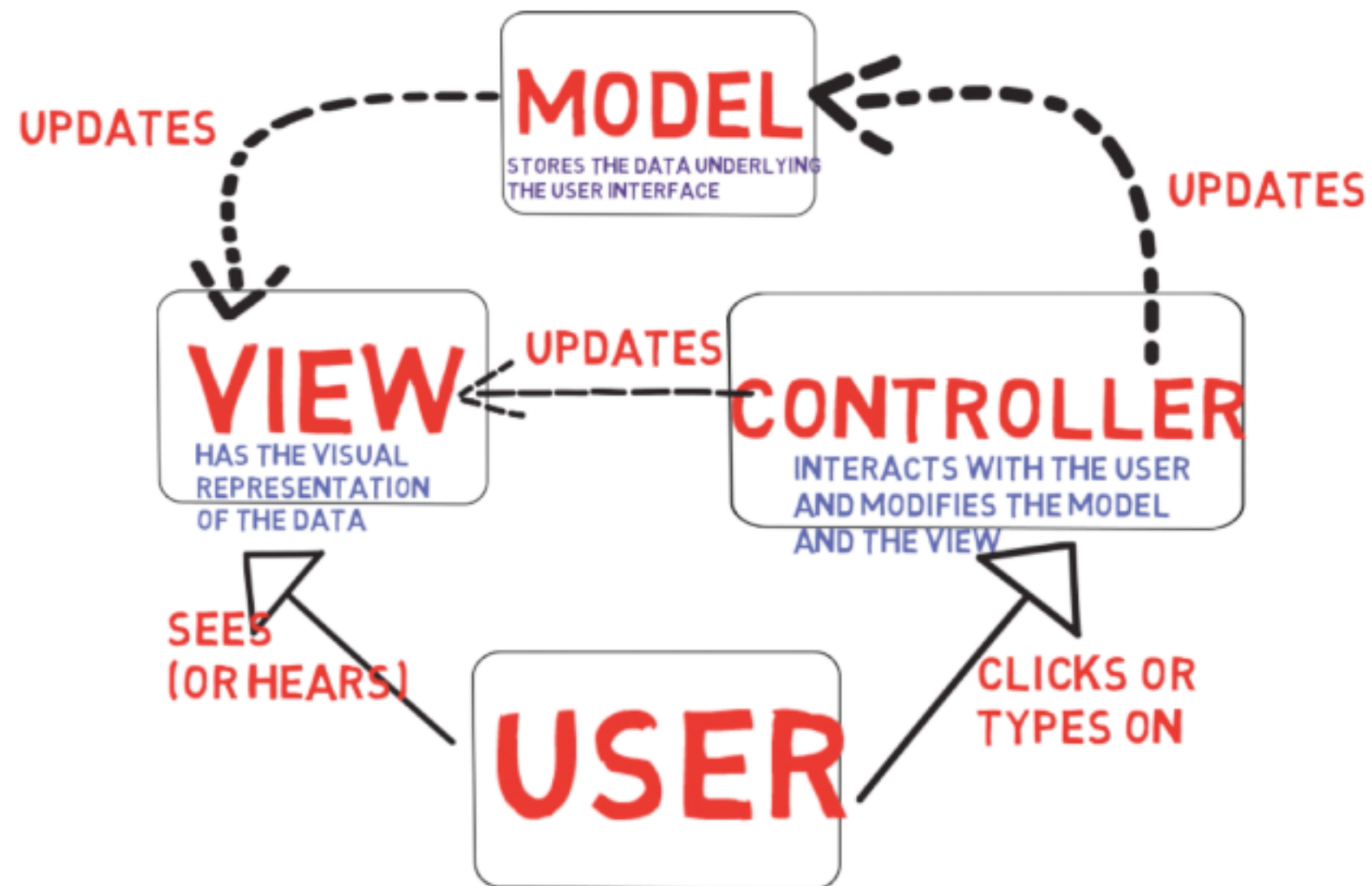AND THE VIEW

SEES
(OR HEARS)

CLICKS OR
TYPES ON

USER

CHARTS AND TABLES

CAN BE UNDERSTOOD VIA THE MVC PARADIGM TOO

BOTH CHARTS AND TABLES ARE ESSENTIALLY

VIEWS
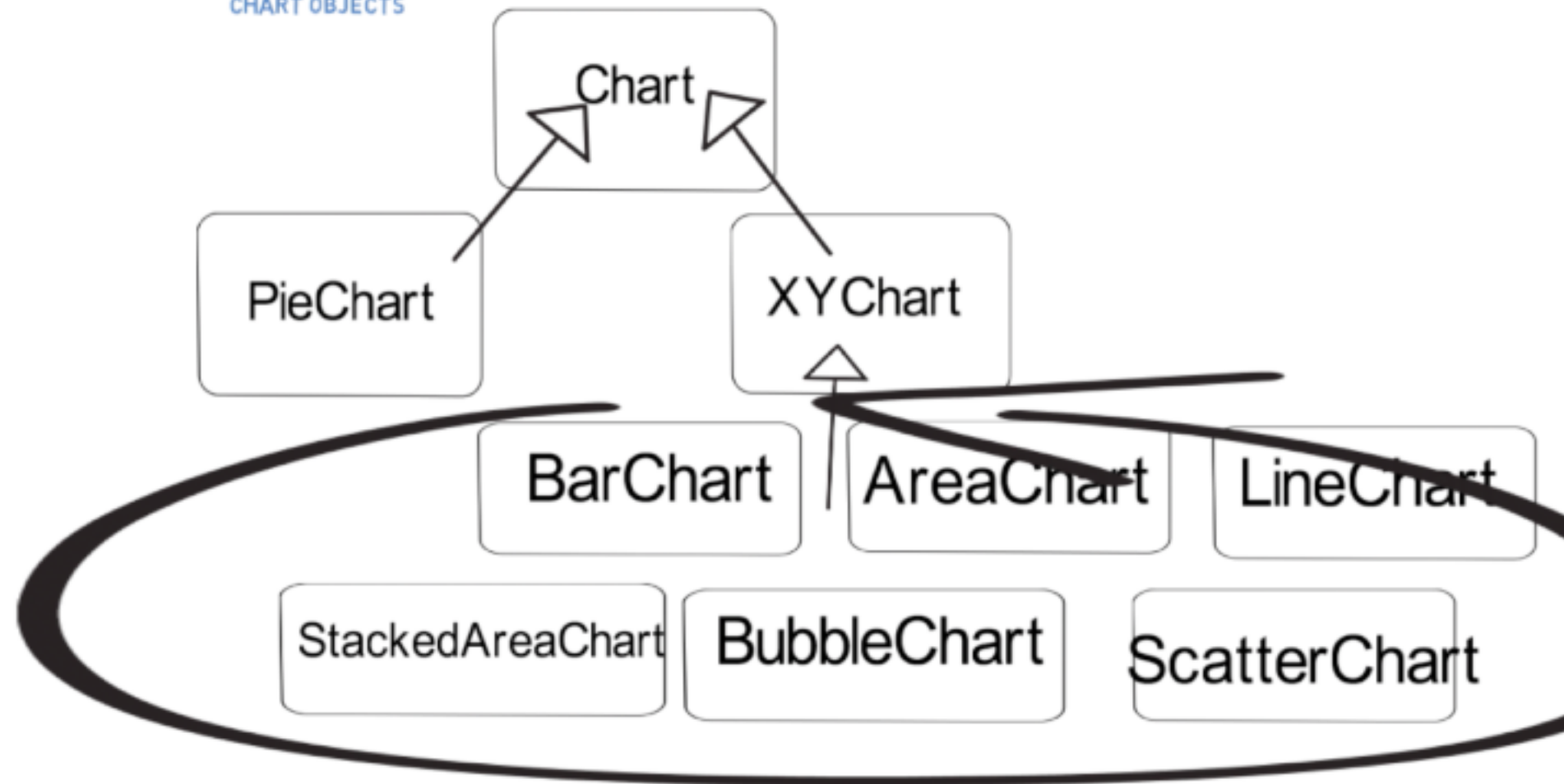
WHERE THE MODEL CONSISTS OF

TABULAR, OFTEN NUMERIC DATA

CHARTS OFTEN DON'T HAVE A CONTROLLER AT ALL

WHY? BECAUSE THE CHART USUALLY IS NOT INTERACTIVE – THE USER DOES NOT INTERACT WITH A CHART

TABLES DO HAVE A CONTROLLER LAYER, AND ITS VERY IMPORTANT WHEN THE MODEL IS EDITABLE

JAVAFX CHARTS

JAVAFX OFFERS A NICE RANGE OF
CHART OBJECTS

Chart

PieChart

XYChart

BarChart

AreaChart

LineChart

StackedAreaChart

BubbleChart

ScatterChart

WIRING MODEL TO VIEW IN CHARTS

A FAIR BIT OF ADAPTER CODE ENDS
UP NEEDING TO BE WRITTEN

```java
final NumberAxis xAxis = new NumberAxis(-5, 70, 7);
final NumberAxis yAxis = new NumberAxis(0,70,7);
xAxis.setLabel("Q2 2015 Revenue, $B");
yAxis.setLabel("Q2 2015 Revenue Y/Y, %");
BubbleChart<Number,Number> bubbleChart = new BubbleChart<~>(xAxis,yAxis);
XYChart.Series<Number, Number> series = new XYChart.Series<>();
for(CompanyMetrics companyMetrics : data) {
    XYChart.Data<Number,Number> dataPoint = new XYChart.Data<~>(companyMetrics.getRevenue(),
            companyMetrics.getGrowthRate(),
            companyMetrics.getMktCap()/100
    );
    series.getData().add(dataPoint);
    StackPane stackPane = new StackPane();
    Text label = new Text(companyMetrics.getName());
    stackPane.getChildren().addAll(label);
    dataPoint.setNode(stackPane);
}

bubbleChart.setLegendVisible(false);
bubbleChart.getData().add(series);
```

WE NEED TO EXPLICITLY
CREATE EACH POINT IN
EACH SERIES OF A GRAPH

BTW, YOU CAN DO ALL KINDS OF
LITTLE NEAT THINGS LIKE CUSTOMIZE
THE LOOK OF EACH POINT

JAVAFX CHARTS ARE YET ANOTHER
EXAMPLE OF THE MVC PARADIGM
AT WORK IN A USER-INTERFACE