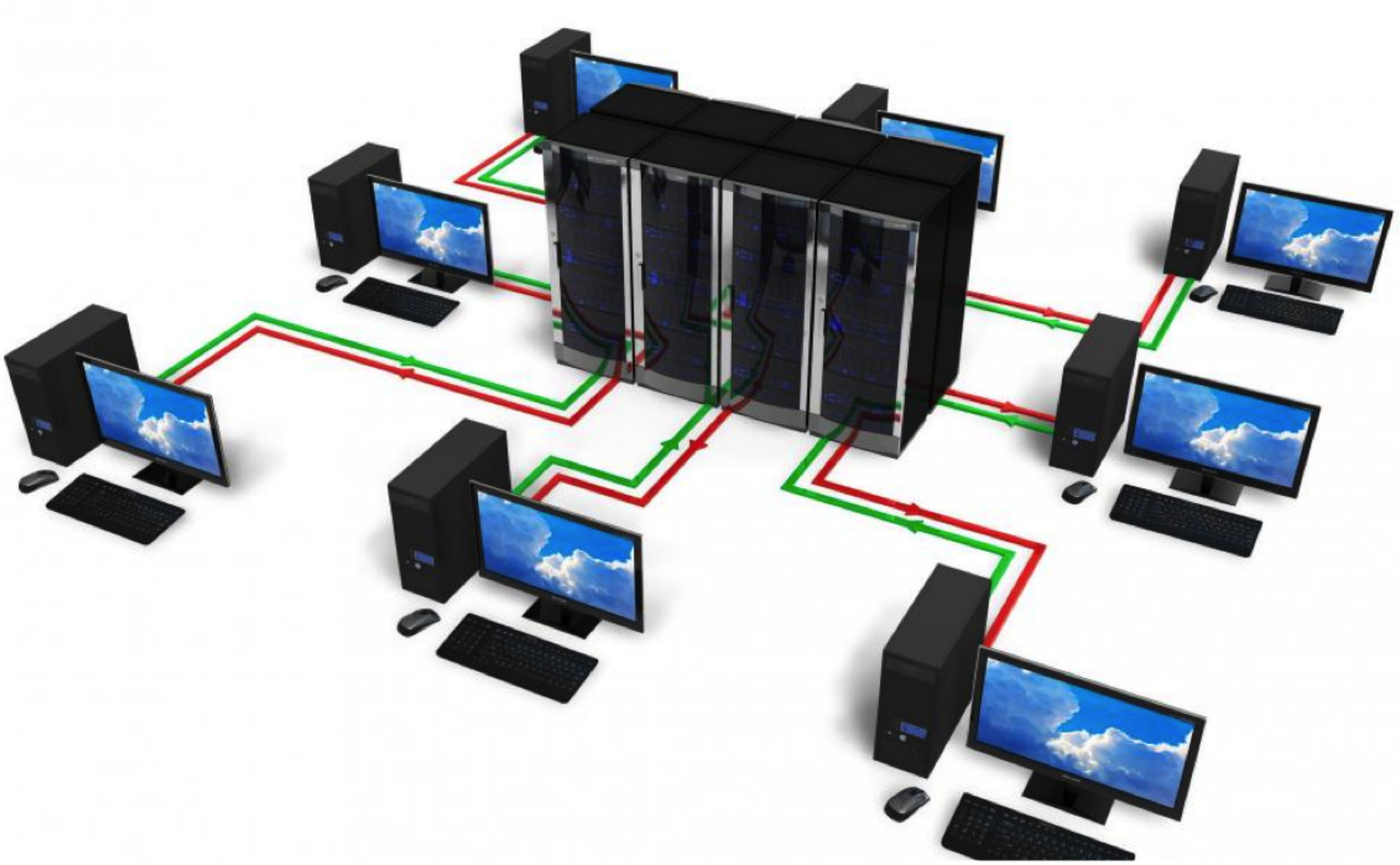
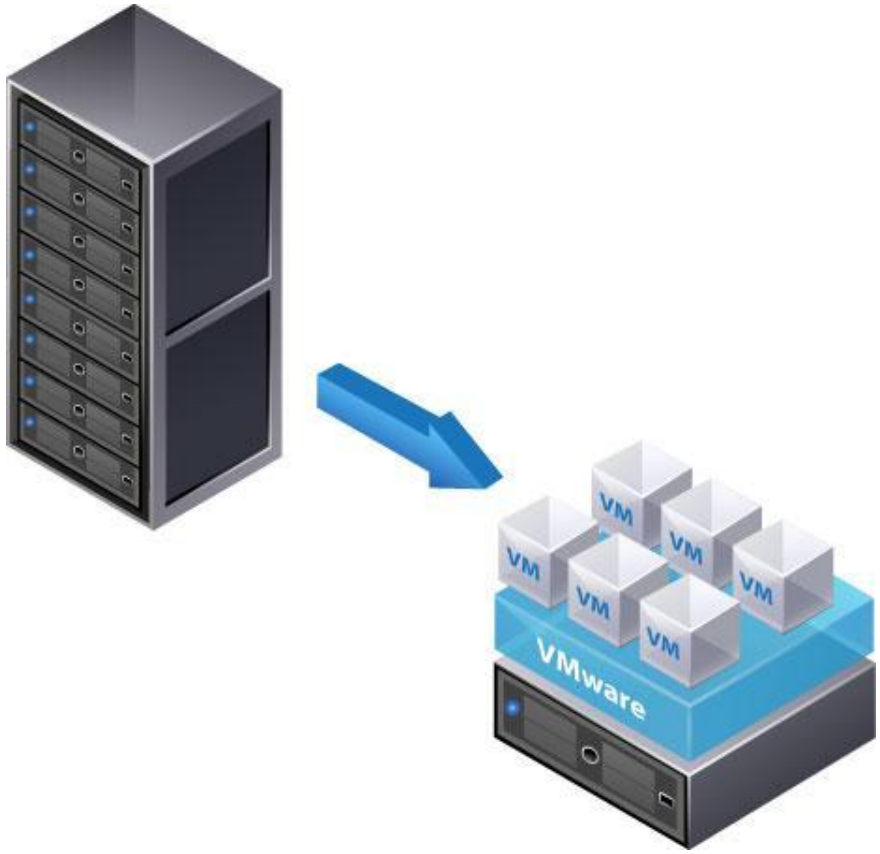
vTimeSeries

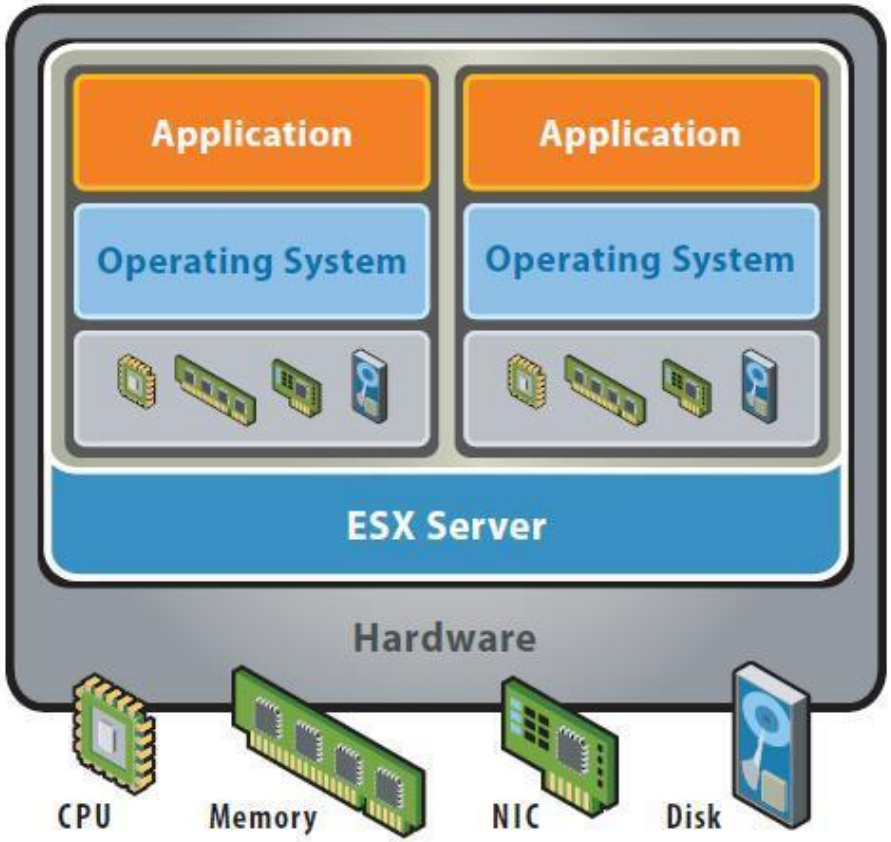
Vladimir Adam, Karan Toor, Cesar Polanco, Ryan McGinley, Nick Cross

Data Center

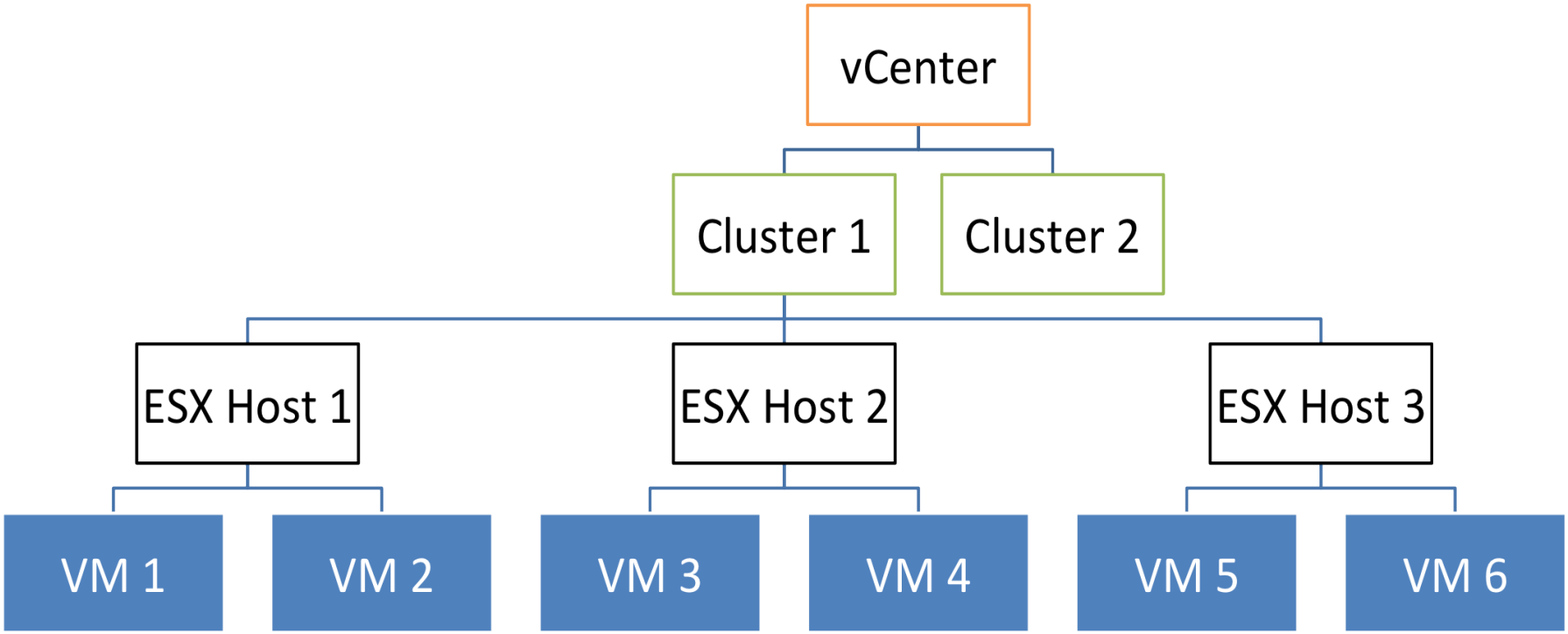


VMware Infrastructure

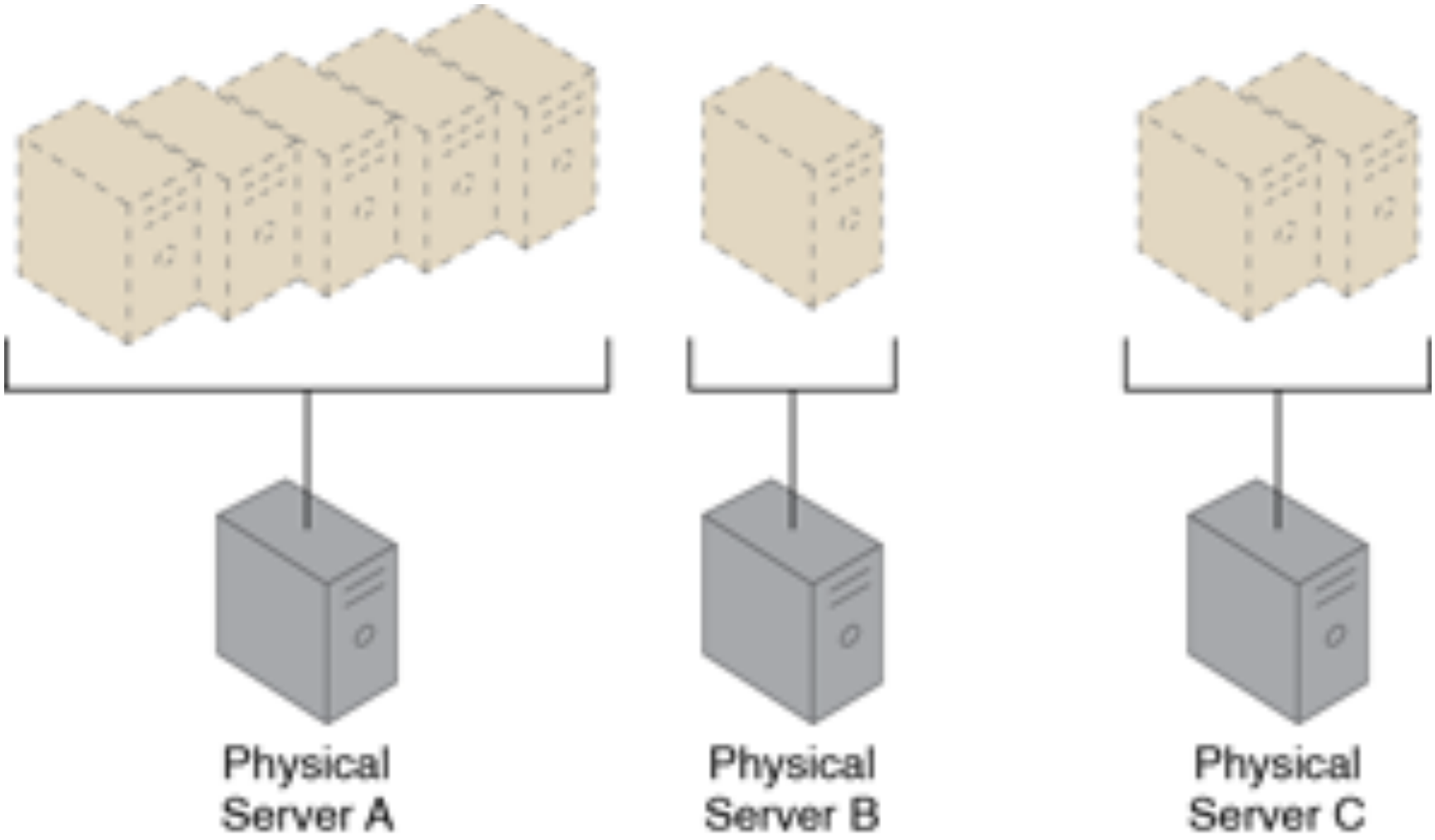




VMware Infrastructure

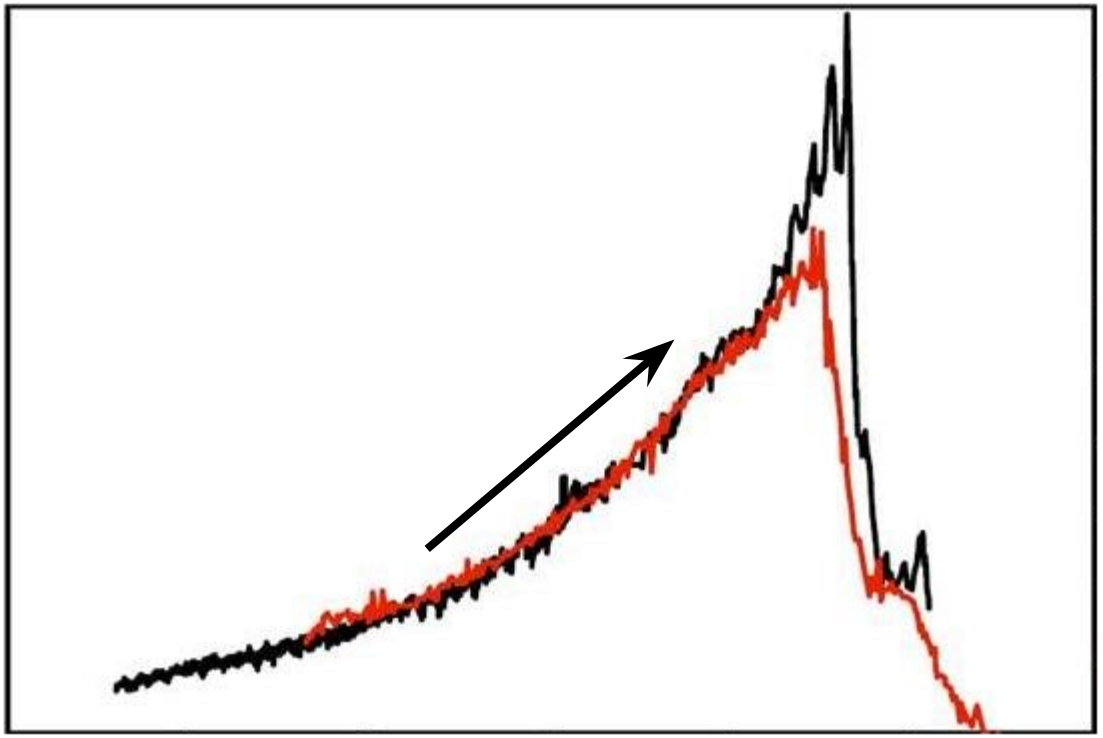


Varying Workload



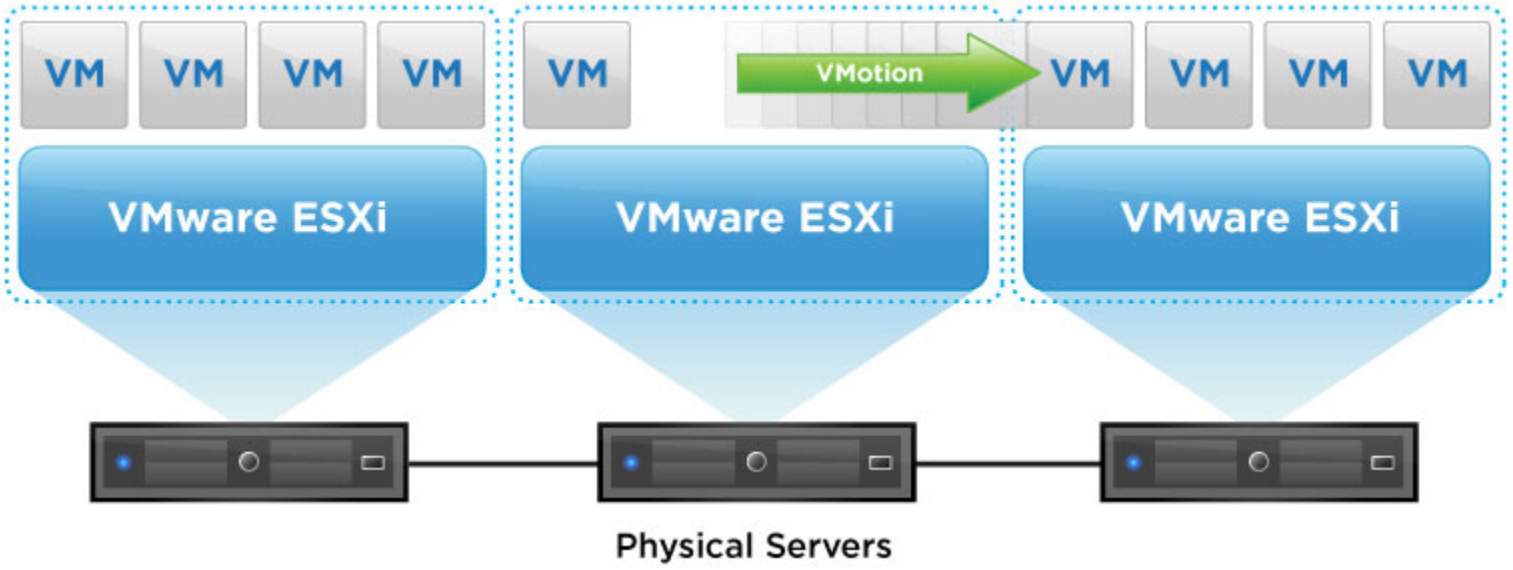
VMware's Current Solution

- DRS (Distributed Resource Scheduler)
- Overload detection based on immediate trend



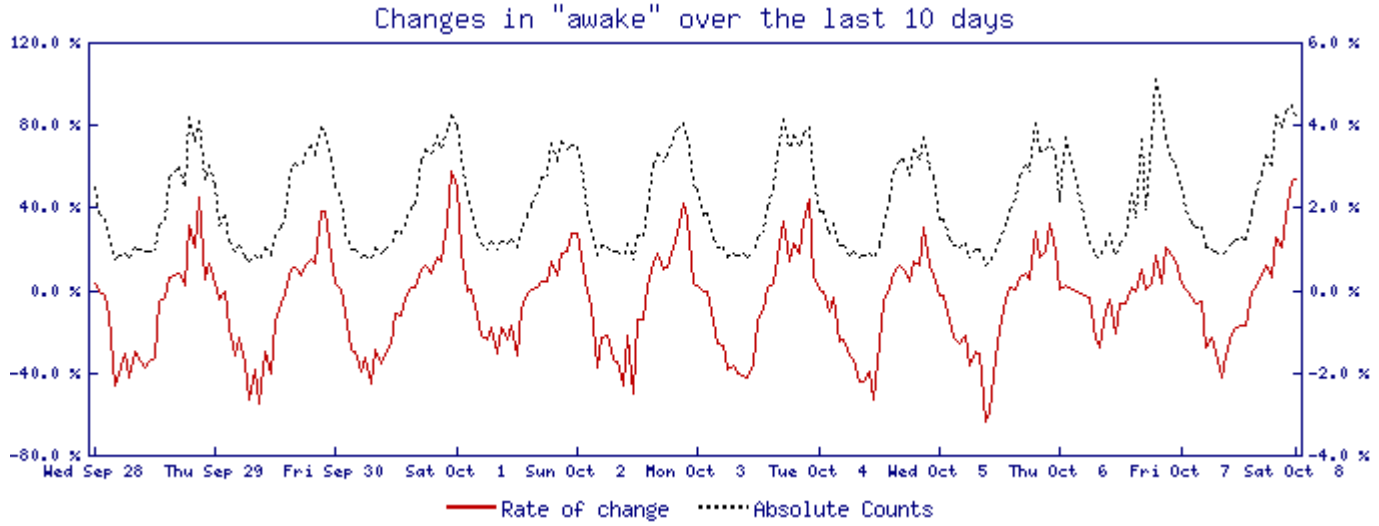
VMware's Current Solution

- VM Migration to less stressed host



Problem

- VM migration is expensive!
- Does not take long term trends into account.



So what's our solution?

Our Solution

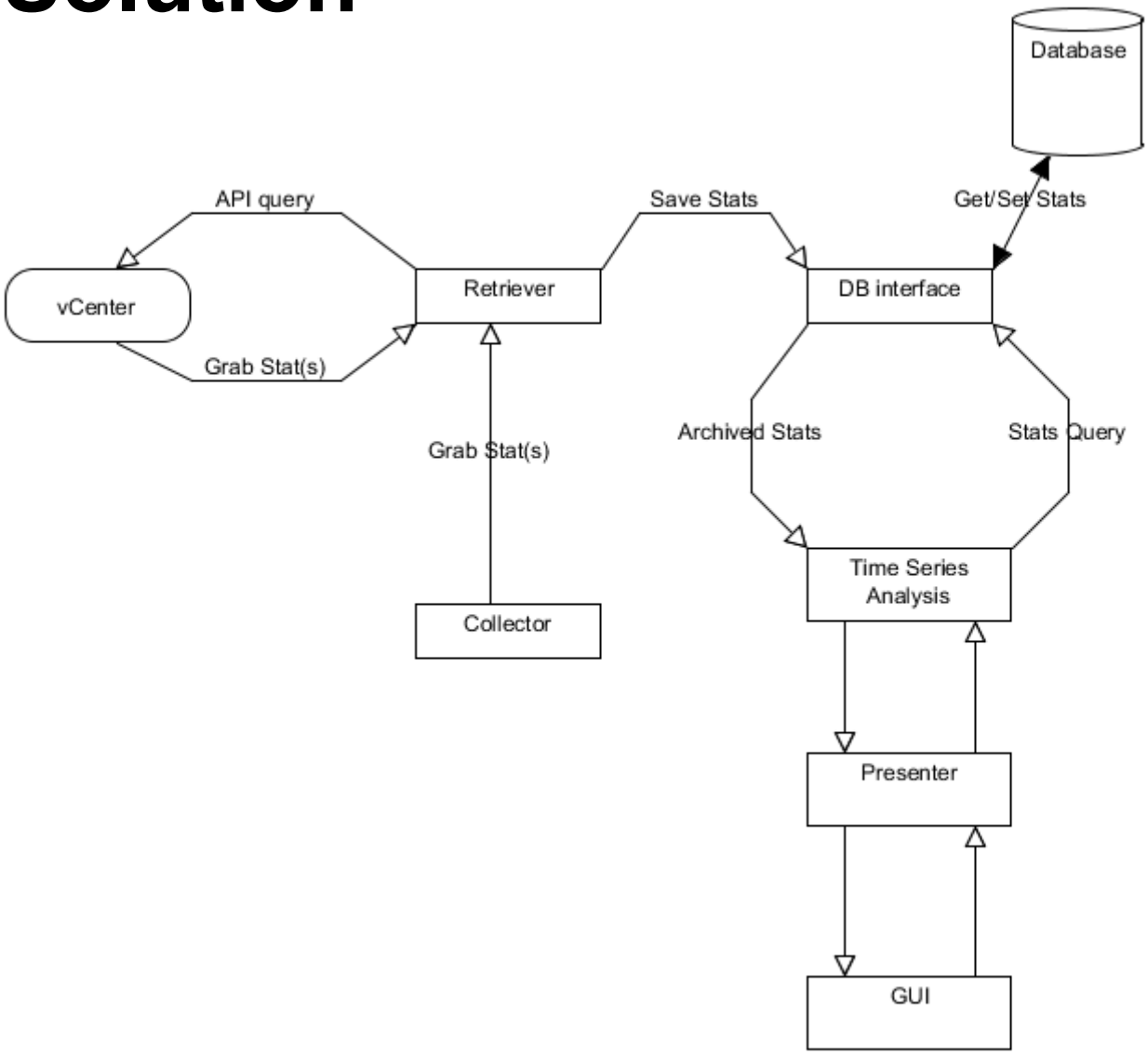
System Administrators Can

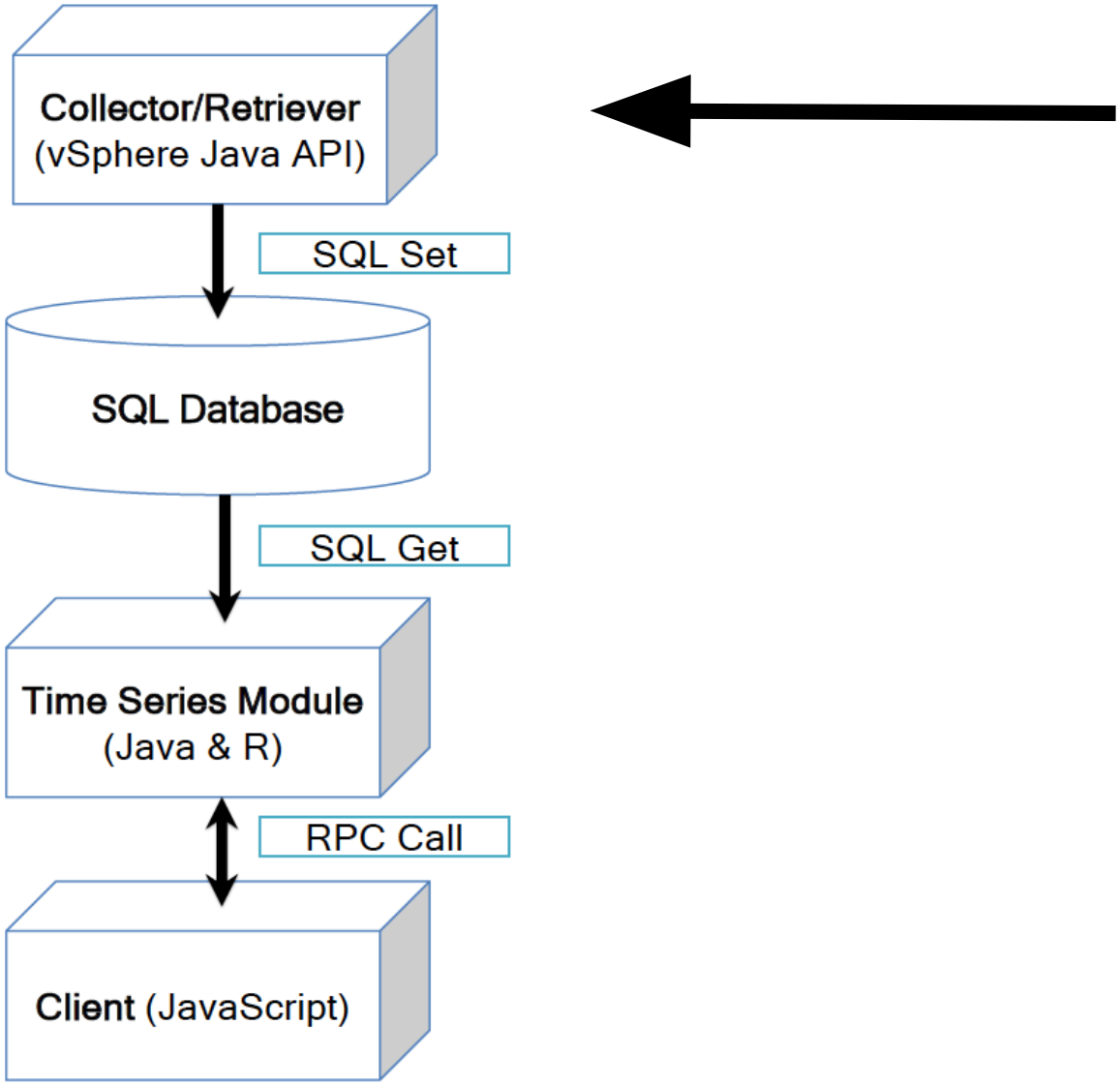
- Forecast hardware demands
- Discover correlations between statistics

Tomorrow	Sat	Sun
		
High: 80% Low: 65%	High: 56% Low: 30%	High: 40mB/s Low: 10mB/s


Network Switch
vCenter Server
NAS
3 ESX Hosts
DevBox

Our Solution




Collector/Retriever
(vSphere Java API)
SQL Set
SQL Database
SQL Get
Time Series Module
(Java & R)
RPC Call
Client (JavaScript)

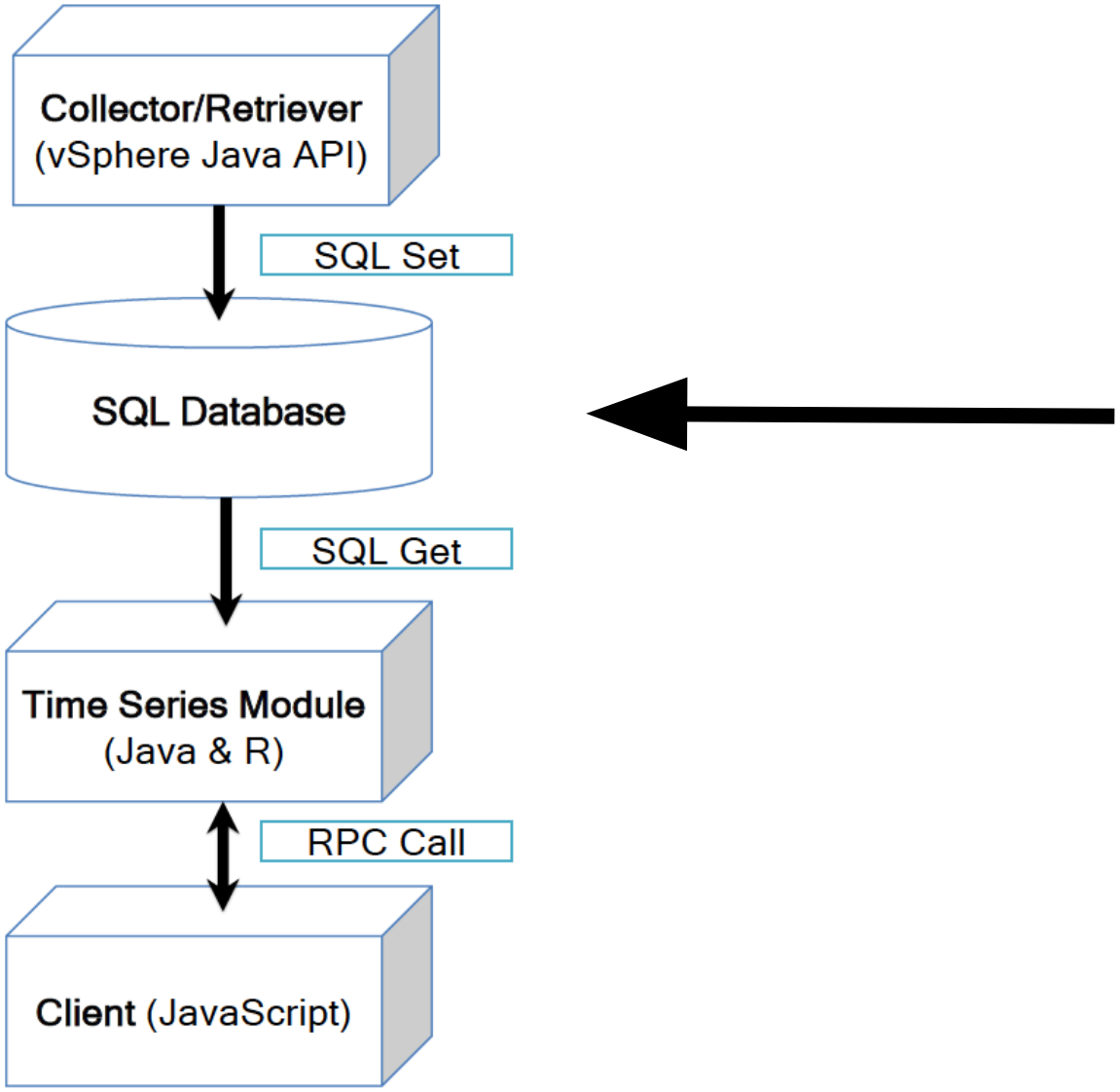
Collector/Retriever

- Use vCenter API to get Data
- Data packed up every 115 seconds

Collector/Retriever

- Use vCenter API to get Data
- Data packed up every 115 seconds

Where does the data go?


Collector/Retriever
(vSphere Java API)
SQL Set
SQL Database
SQL Get
Time Series Module
(Java & R)
RPC Call
Client (JavaScript)

Database

MySQL DB

- Created objects for both input and output

Database

MySQL DB

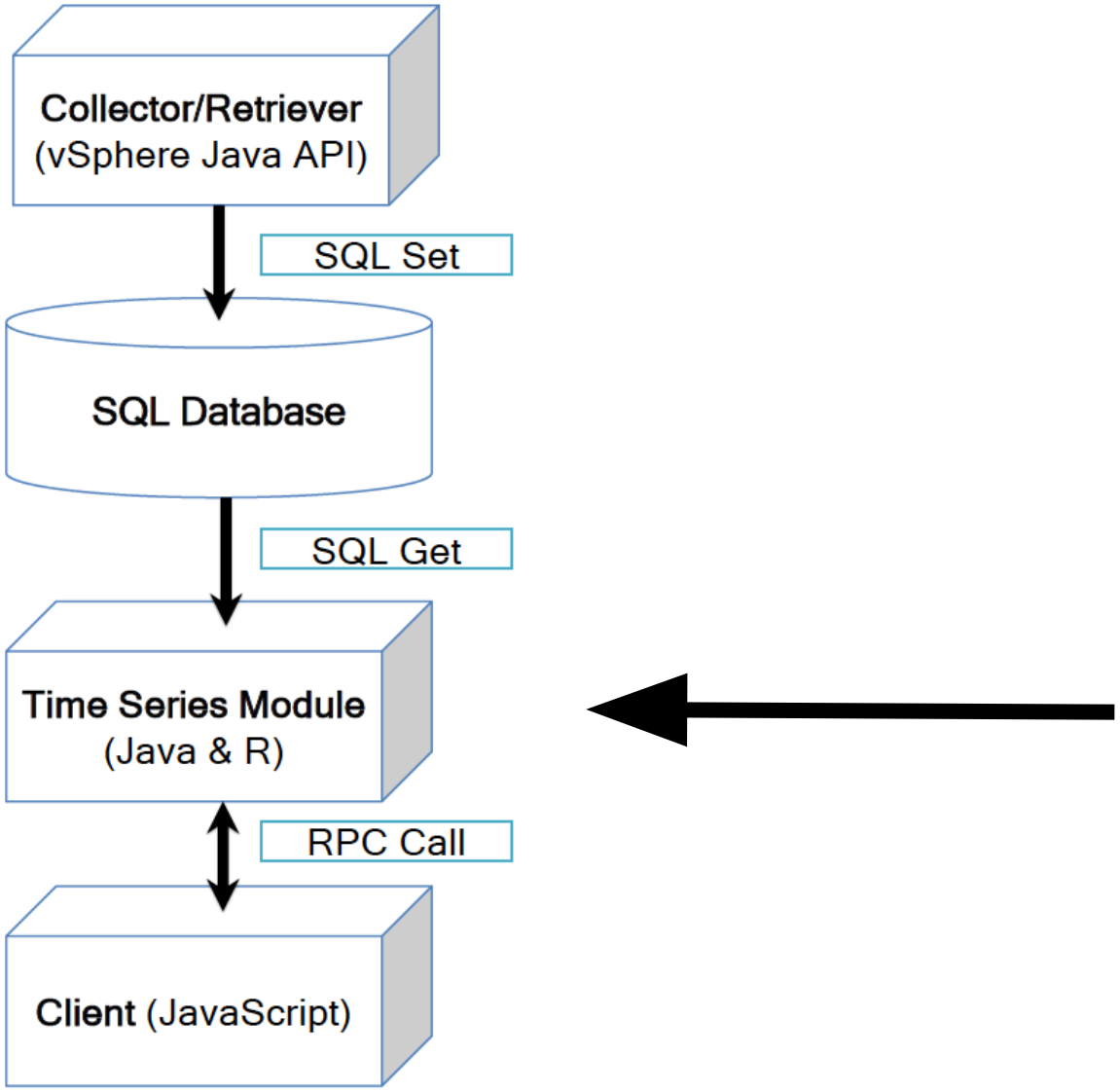
- Created objects for both input and output
- Using database created some conflicts
  - Naming conventions
  - Creating tables/Entities

Database

MySQL DB

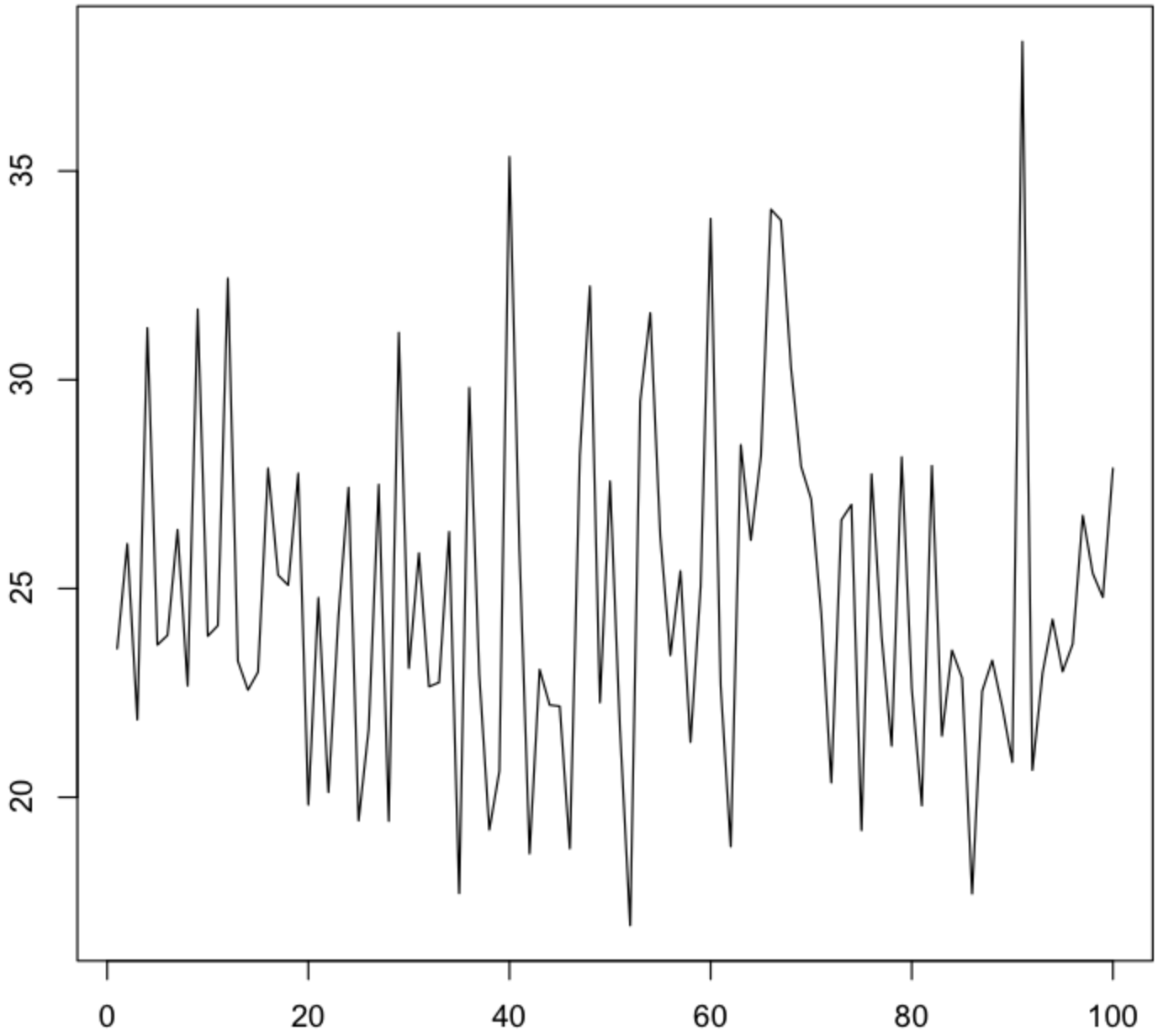
- Created objects for both input and output
- Using database created some conflicts
  - Naming conventions
  - Creating tables/Entities

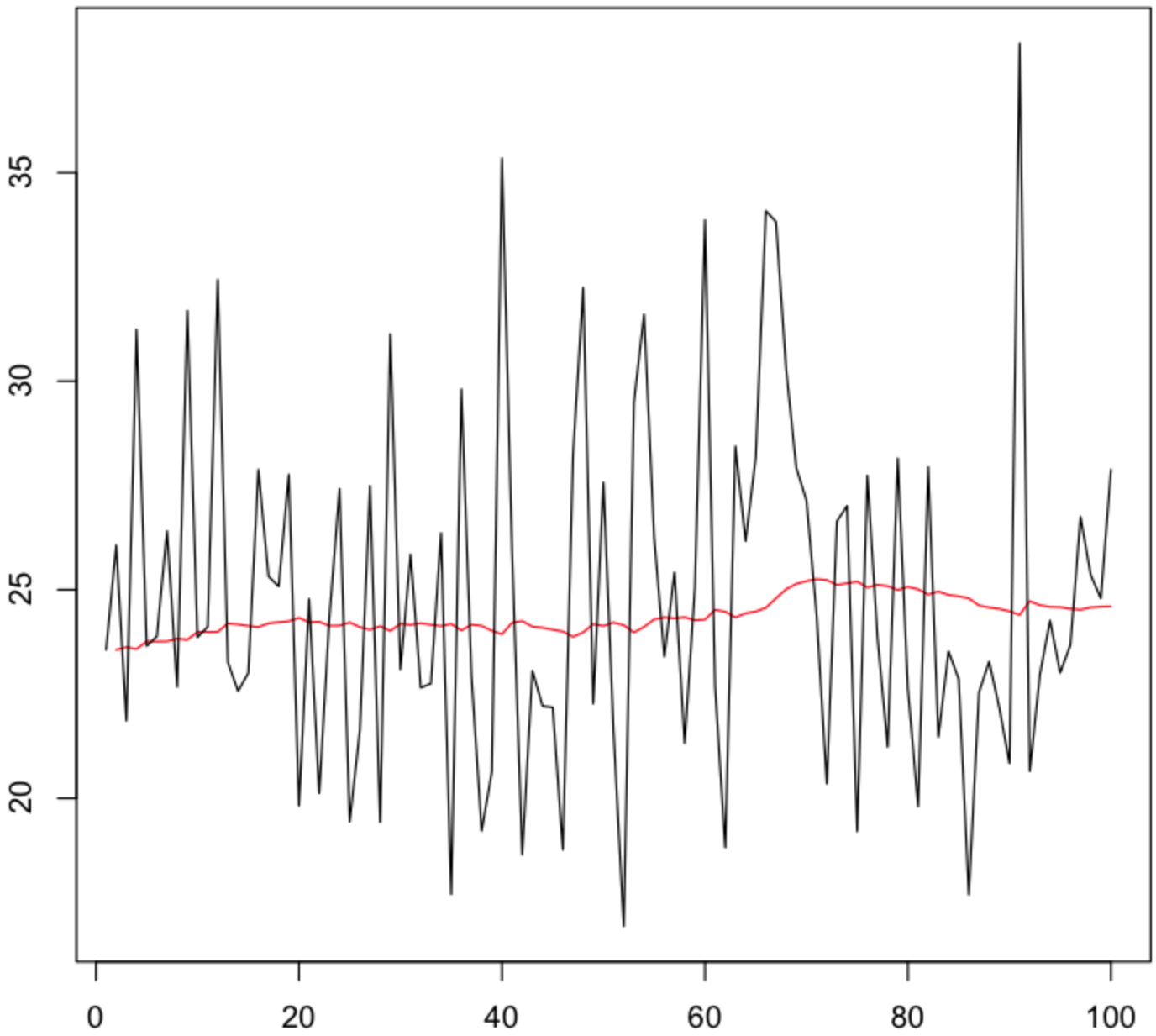
Where does the analysis take place?


Collector/Retriever
(vSphere Java API)
SQL Set
SQL Database
SQL Get
Time Series Module
(Java & R)
RPC Call
Client (JavaScript)

Single Exponential Smoothing

$$s_t = (1-\alpha)^{t-1} x_0 + \alpha \sum_{i=1}^{t-1} (1-\alpha)^{i-1} x_{t-i}$$


35
30
25
20
0
20
40
60
80
100

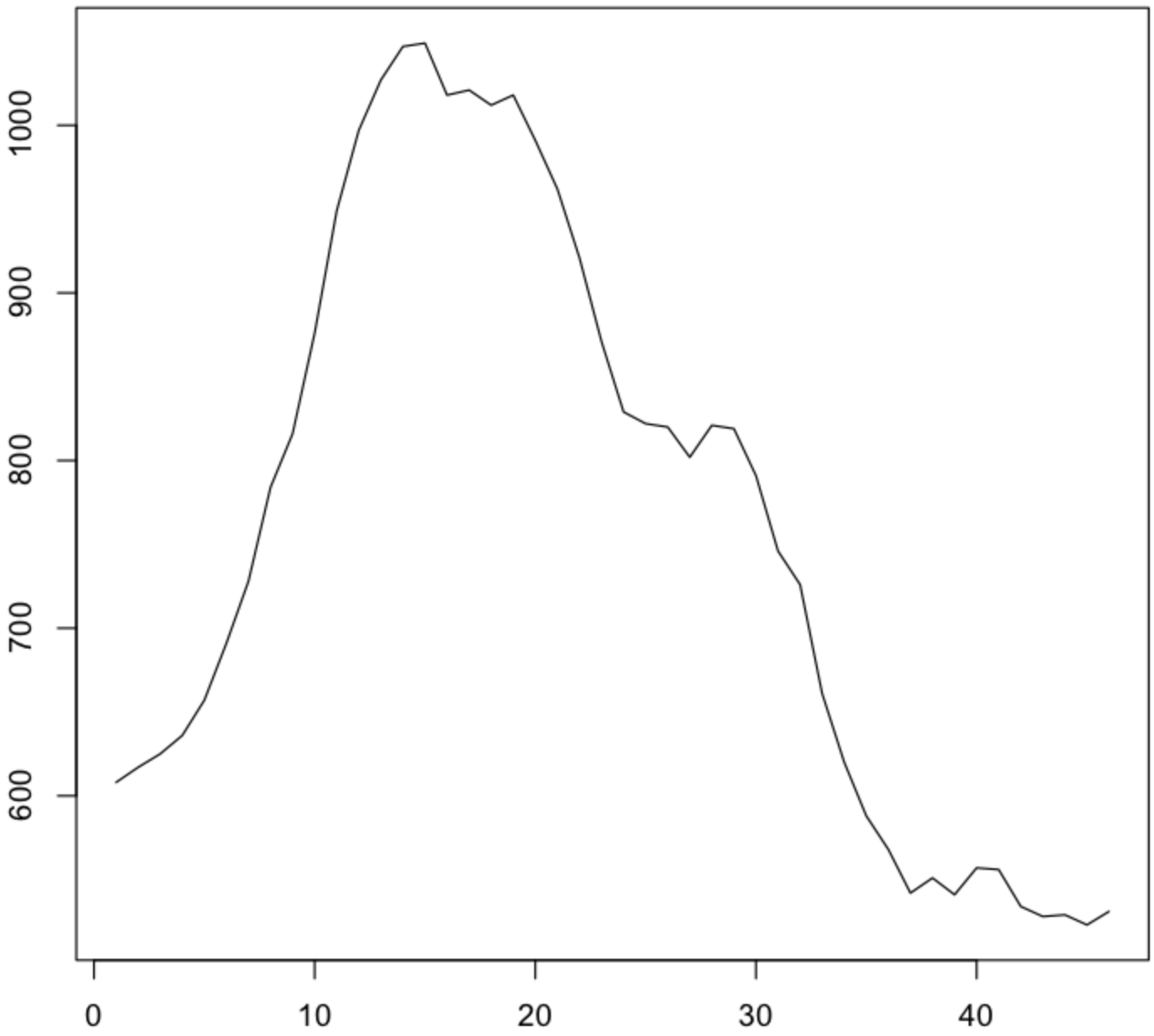

35
30
25
20
0
20
40
60
80
100

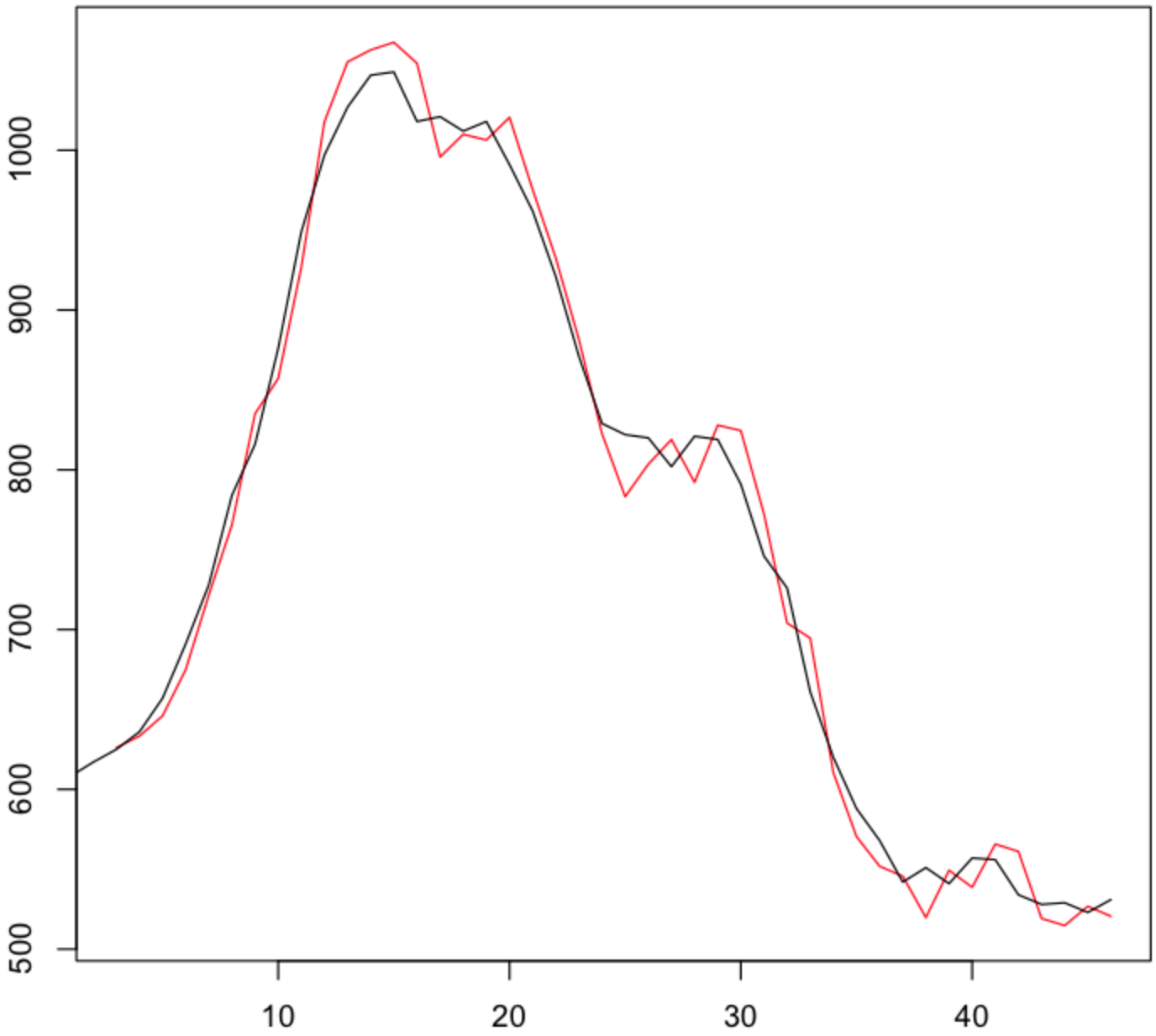
Double Exponential Smoothing

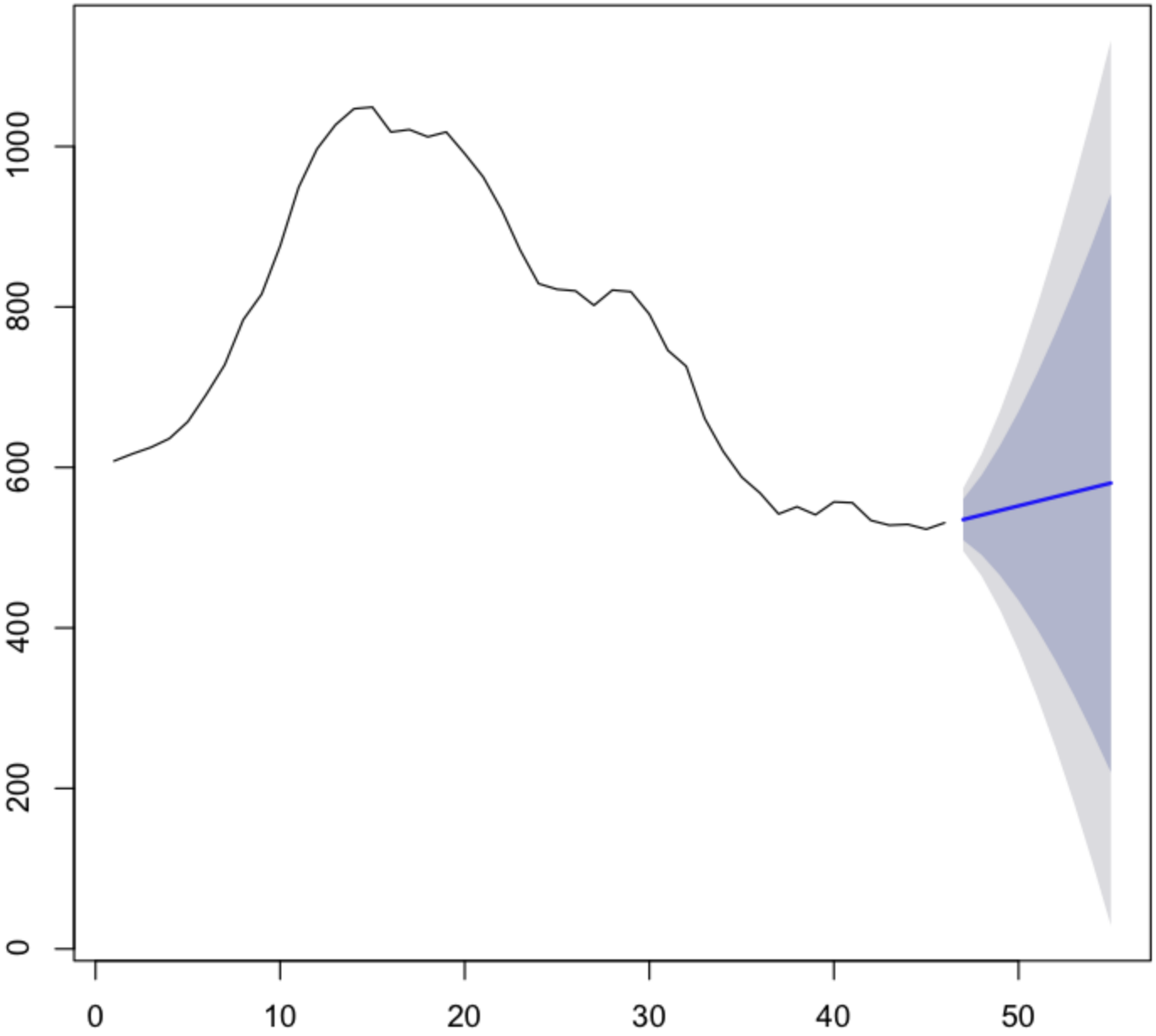
$$s_t = \alpha \sum_{i=0}^{t-1} (1-\alpha)^i (x_{t-i} + b_{t-i-1})$$

$$b_t = \beta \sum_{i=0}^{t-2} (1-\beta)^i (s_{t-i} - s_{t-i-1})$$

$$F_{t+m} = s_t + m b_t$$


1000
900
800
700
600
0
10
20
30
40


1000
900
800
700
600
500
10
20
30
40


0
200
400
600
800
1000
0
10
20
30
40
50

Triple Exponential Smoothing

$$s_t = \alpha \frac{x_t}{c_t - L} + \sum_{i=0}^{t-1} (1-\alpha)^i (x_{t-i} + b_{t-i})$$

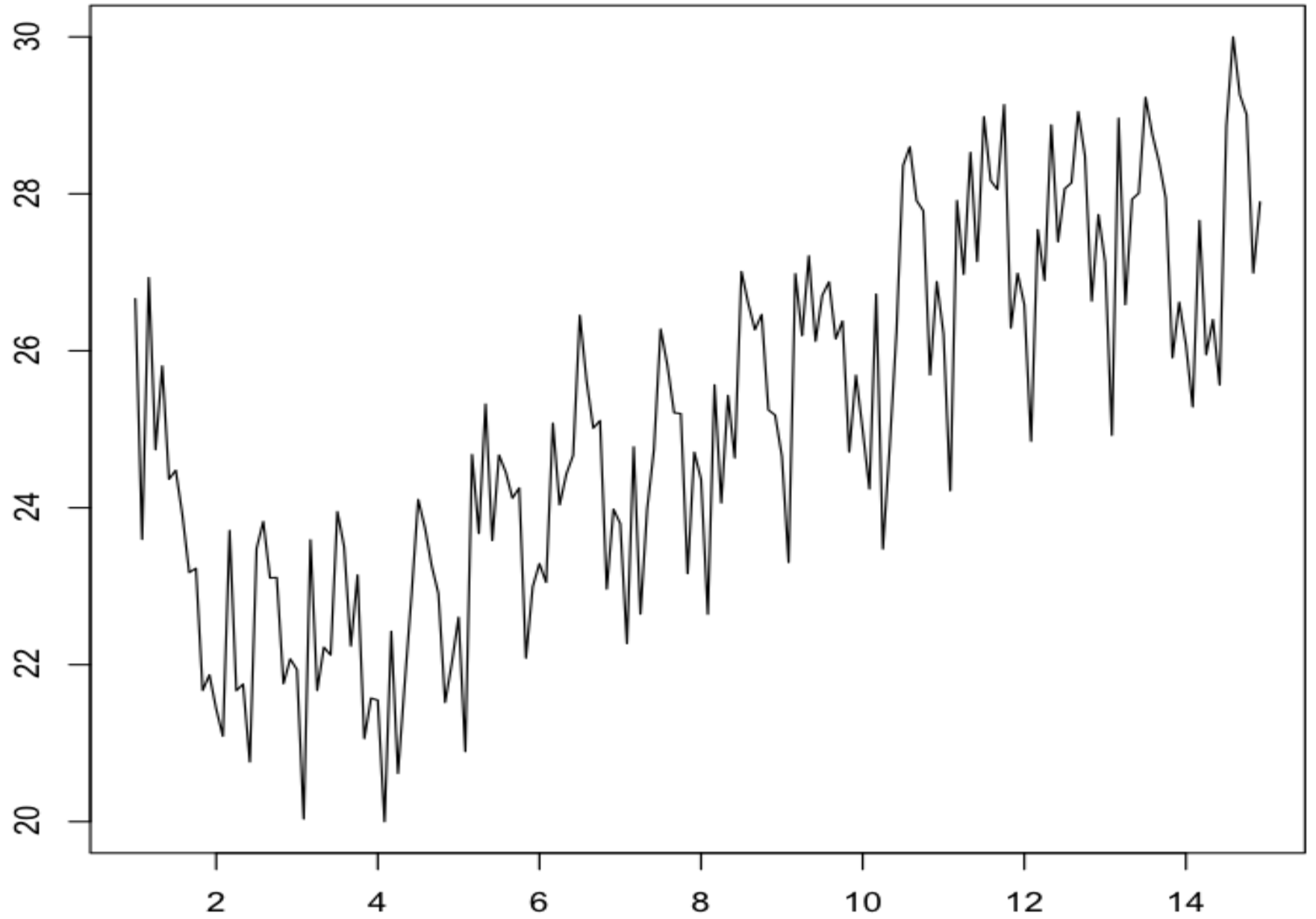
$$c_i = \frac{1}{N} \sum_{j=1}^{N} \frac{x_{L(j-1)i}}{A_j}$$

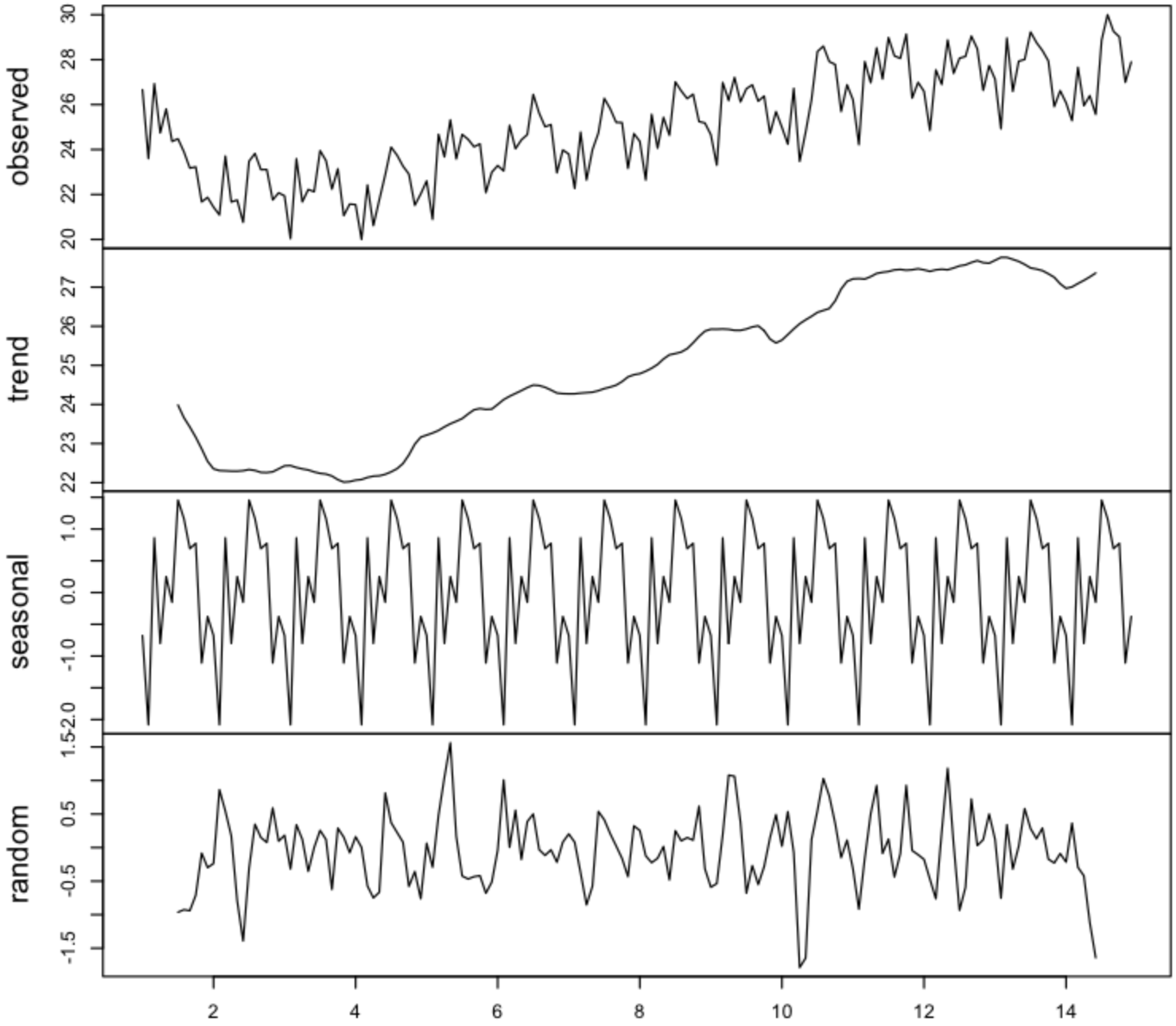
$$b_t = \beta \sum_{i=0}^{t-2} (1-\beta)^i (s_{t-i} - s_{t-i-1})$$

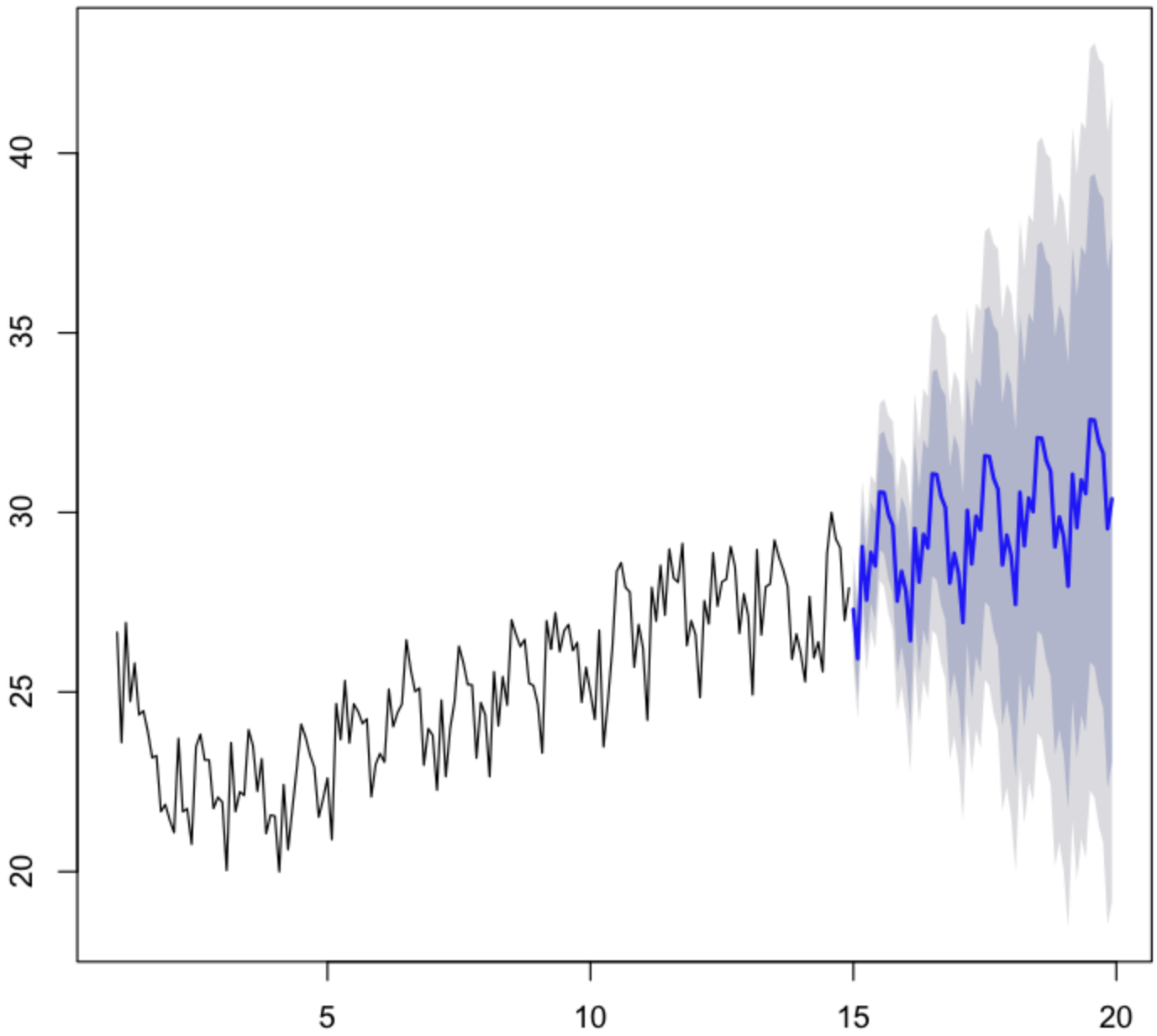
$$A_j = \frac{1}{L} \sum_{i=1}^{L} x_{L(j-1)+i}$$

$$c_t = \gamma \frac{x_t}{s_t} + (1-\gamma) c_{t-L}$$

$$F_{t+m} = (s_t + m b_t) c_{t-L+((m-1) \mod L)}$$


30
28
26
24
22
20
2
4
6
8
10
12
14

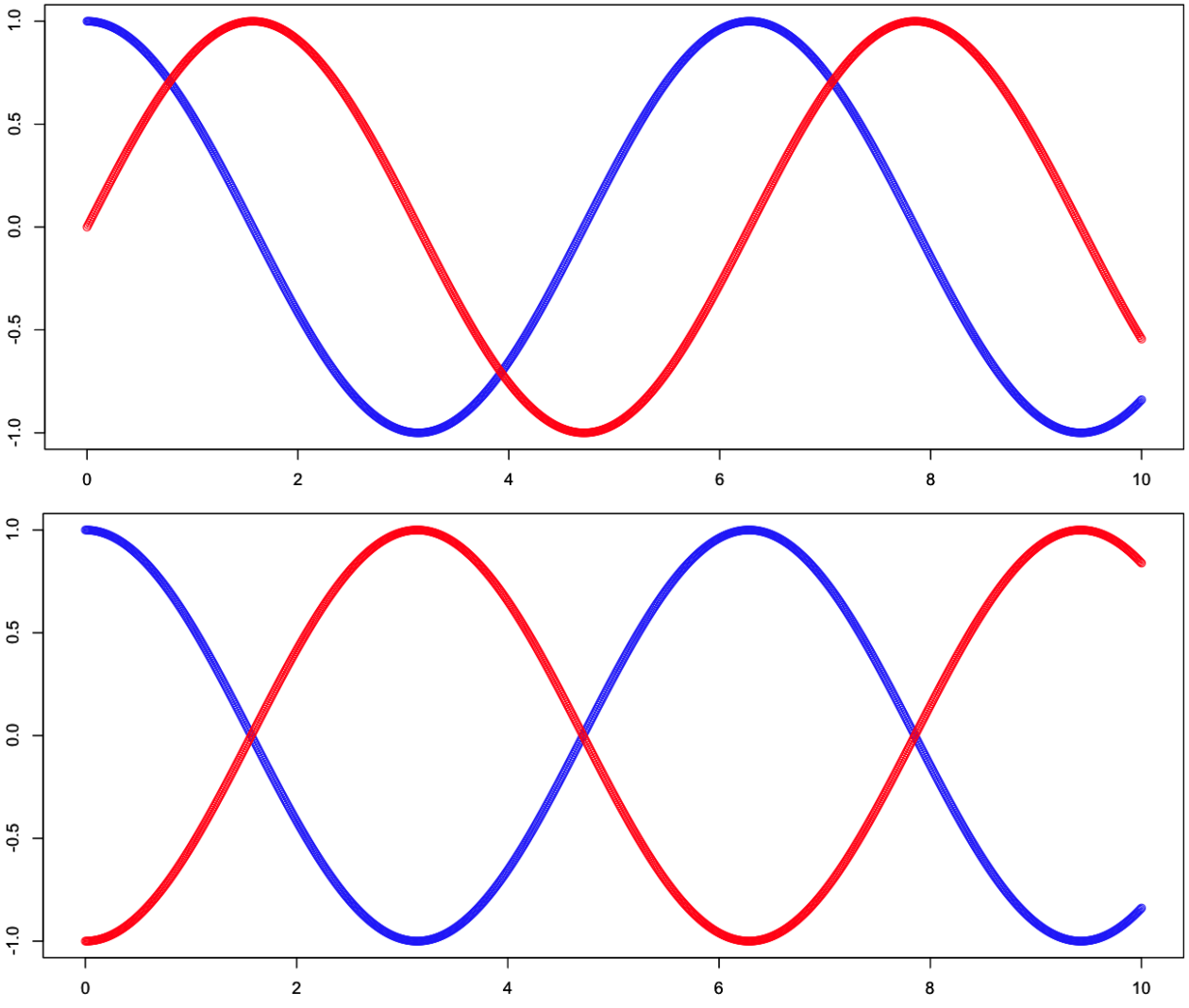

observed
trend
seasonal
random
20
22
24
26
28
30
22
23
24
25
26
27
-2.0
-1.0
0.0
1.0
1.5
-1.5
-0.5
0.5
2
4
6
8
10
12
14

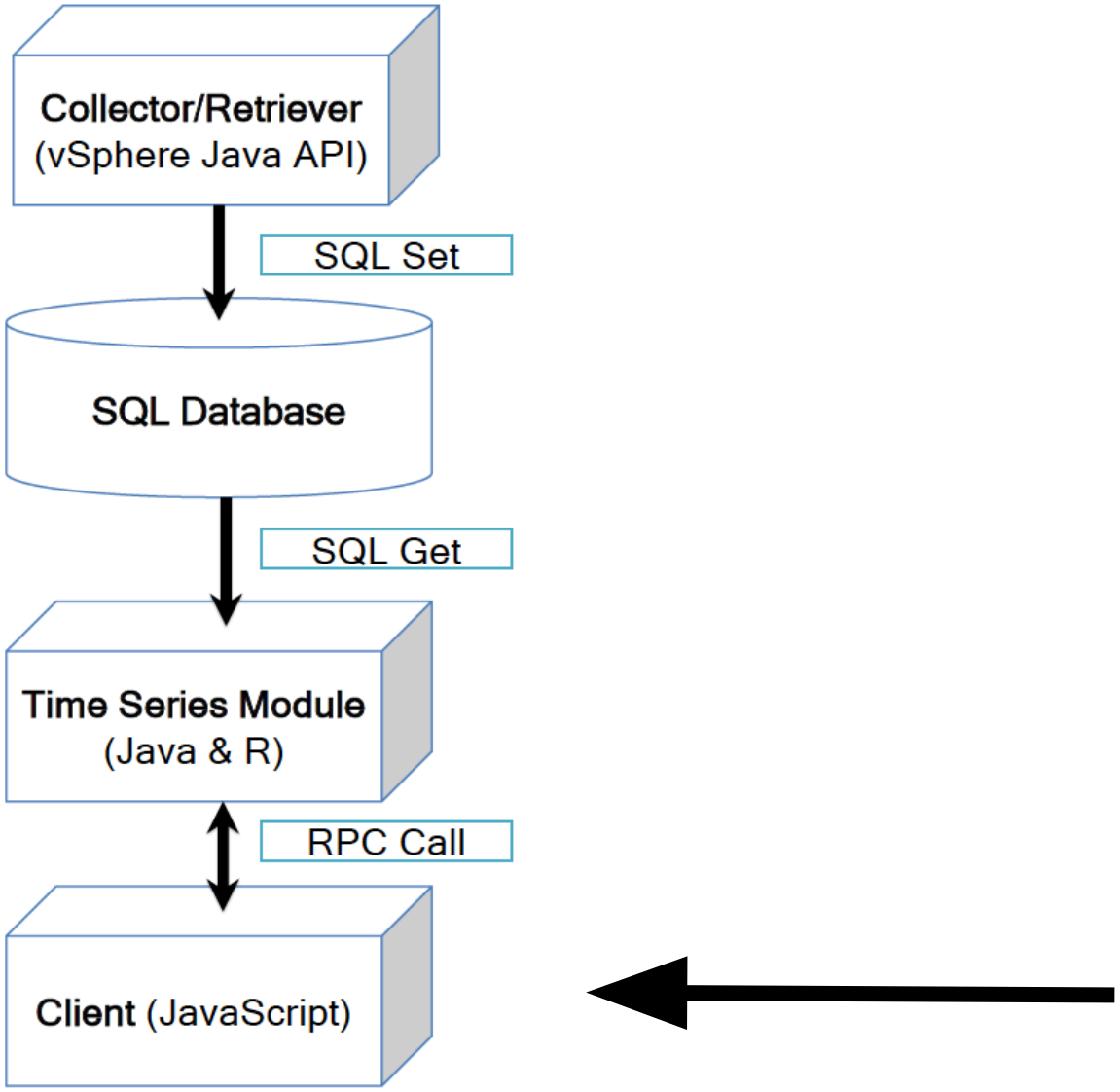

40
35
30
25
20
5
10
15
20

Correlation Coefficient

$$\rho_{xy}(\tau) = \frac{\frac{1}{n}\sum(X_t - \mu_x)(Y_{t+\tau} - \mu_y)}{\sigma_x \sigma_y}$$

Correlation Among Stats

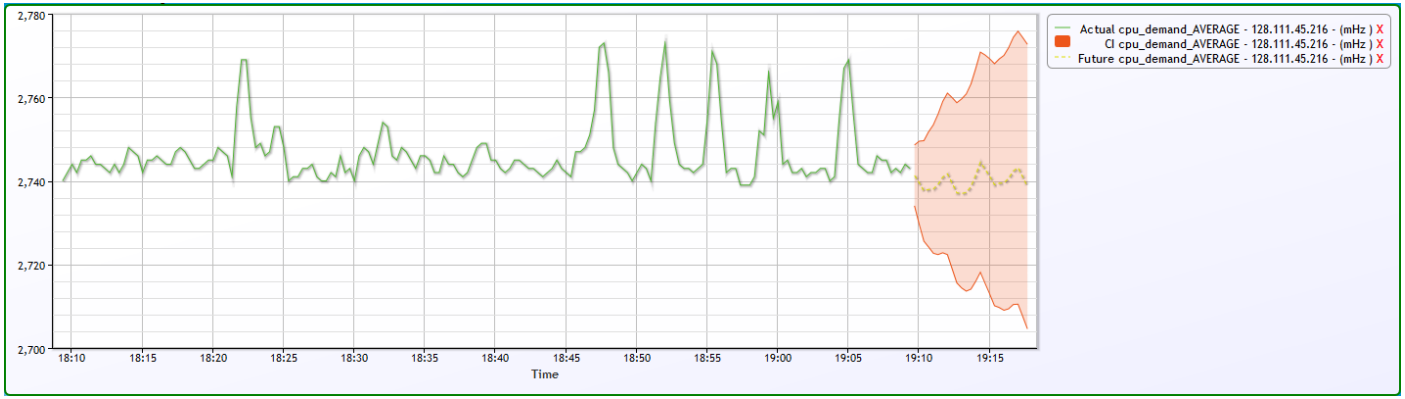



Collector/Retriever
(vSphere Java API)
SQL Set
SQL Database
SQL Get
Time Series Module
(Java & R)
RPC Call
Client (JavaScript)

Front-end

- Remote Procedure Calls
- Google Web Toolkit

- Highcharts



Demo

Conclusion

- Forecast a typical day (e.g. Mon, Tue, etc..)

- Do automated scheduling based on information