Pipes and Filters

Alina Rozenbaum

Problem

We need to build a system:

- That must be built by several people
- For which the steps must be individual, yet highly cohesive
- That can change along with the requirements for it

Context

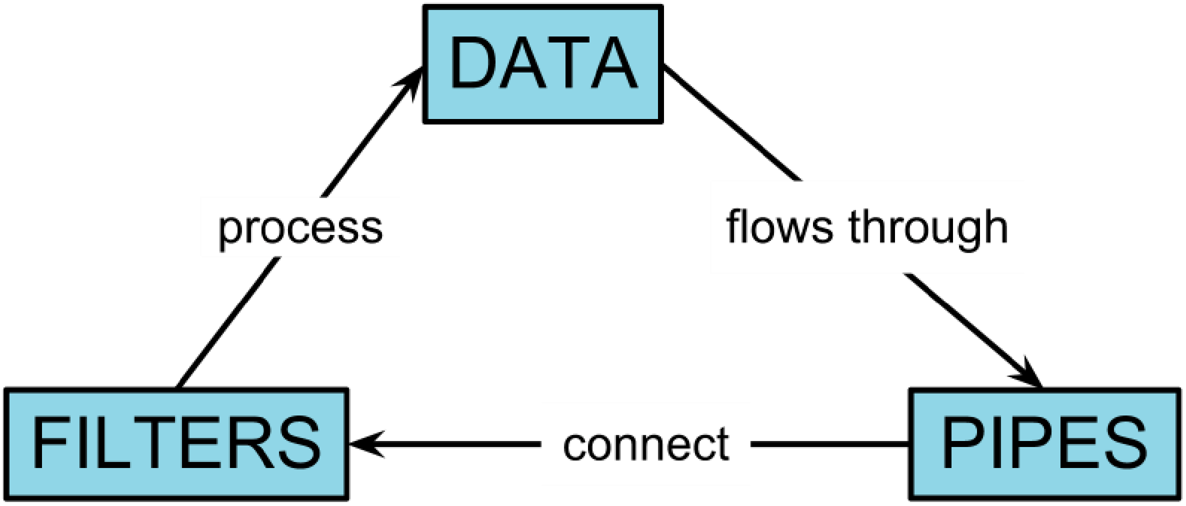
There must be programs that will process streams of data.

Forces

The following forces must be addressed:

- Separate filters should be interchangeable, replaceable and moveable
- Non-adjacent steps share no info
- Can execute parallel steps
- Can display/store final results many ways
- Various sources of data input
- Small and many rather than big and few
- No more than one action per step

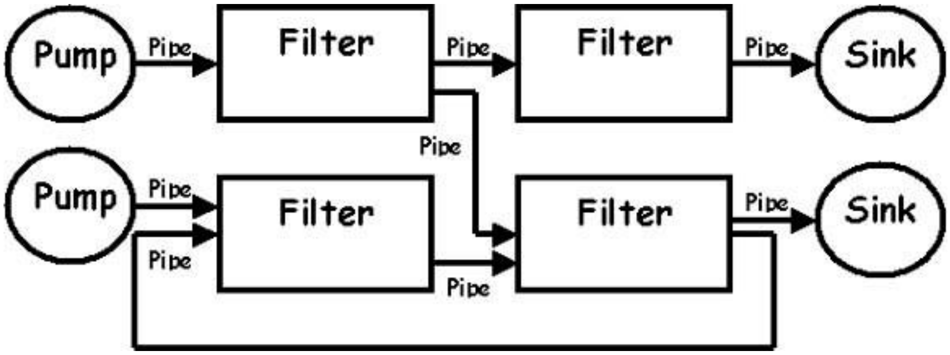
Solution



- Divide into processing steps
- Let each step be a filter that takes input data, acts on it, and produces the output data for the final product, or the next filter.
- Connect the outputs, one after another in a sequential line.
- Make sure that filters can act in parallel if need be.
- The input should be a data source, such as a file.
- The output should be a data sink, such as a display or write-up of data.

Resulting Context

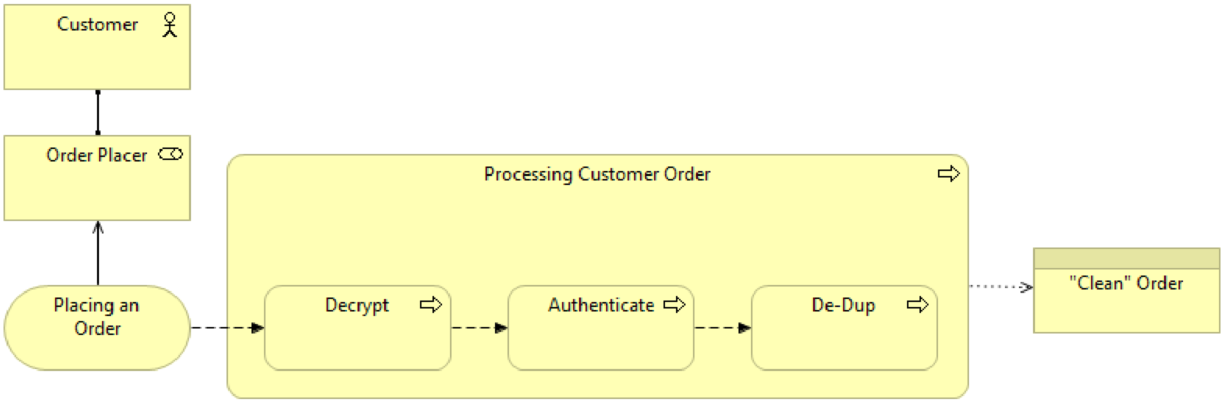
> *"The Pipes and Filters architectural pattern provides a structure for systems that process a stream of data. Each processing step is encapsulated in a filter component. Data [are] passed through pipes between adjacent filters. Recombining filters allows you to build families of related filters." [Buschmann]*(2)



Implementation

1. Divide the problem
2. Define data type and format
3. Decide how to make pipe connections
4. Design and implement the filters
5. Decide how to handle errors
6. Configure the system and run it

Examples



Here we have an example of an order being placed at a company website by a customer, and it going through the preliminary processing within the company.

Once the customer places the order, the code must be decrypted. Then, it must be verified or authenticated as a valid order, from a valid customer. Finally, if the customer, say pressed the button too many times for “Buy” on the page, and duplicate orders were sent, they are weeded out. The result is a “clean” order that the company can now fill.

Known Uses

Data processing and flow.

References

(1): http://www.csee.wvu.edu/~ammar/CU/swarch/lecture%20slides/slides%204%20sw%20arch%20styles/supporting%20slides/SWArch-4-PipesandFilter.pdf

(2): http://www.cs.olemiss.edu/~hcc/csci581oo/notes/pipes.html

(3): http://msdn.microsoft.com/en-us/library/ee658117.aspx

(4): http://pubs.opengroup.org/architecture/togaf9-doc/m/chap25.html

(5): http://enterpriseintegrationpatterns.com/PipesAndFilters.html

(6): http://www.dossier-andreas.net/software_architecture/pipe_and_filter.html

(7): http://teaching.software-carpentry.org/2012/09/06/week-1-shell-pipes-and-filters/

(8): http://i.msdn.microsoft.com/dynimg/IC22368.gif