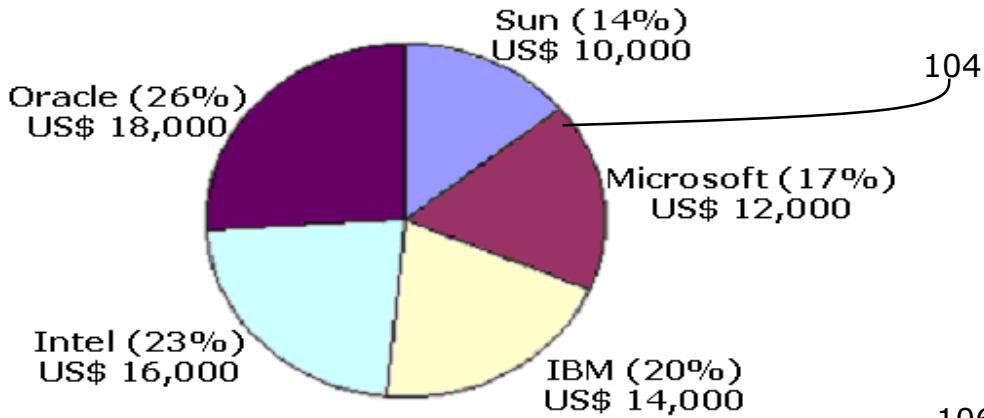


A simple SCC for presenting a Pie chart for portfolio allocations



Total value of the investment is: US\$ 70,000

Pseudo-code for including and presenting a simple Pie-chart SCC in an application, for a portfolio of a given account number

1. /* Below is application code for constructing simple non-replaceable-SCC:-
2. A Pie-chart for showing an investment portfolio of a given account-number*/
3. // Instantiate an object instance using a reusable Pie-chart class:
4. GUI_PieChart Pie1 = new GUI_PieChart(Application_Context_Object);
- 5.
6. // The application must Implement application code to access data for the
7. // account number and process the data for constructing/configuring Pie chart.
8. /*- Application code to get data is not shown to keep pseudo-code brief, -*/
- 9.
10. // Assume that the data obtained by above application code is as follows:
11. Double val[] = { 10000, 12000, 14000, 16000, 18000 };
12. String names [] = {"Sun", "Microsoft", "IBM", "Intel", "Oracle"};
13. String Title = "Total value of the investment is: US\$ 70,000";
- 14.
15. // Use the application data for initialize/configure the Pie chart object
16. Pie1.setData(5, val, names, Title);
17. /*- Above completely initialized/configured Pie chart object is a SCC -*/
18. // The following statement adds the SCC to a parent component for display
19. ParentDrawingArea.AddChild (Pie1, x_location, y_location, null);
20. /* It requires removing whole application code above, implemented for assembling the SCC, must be removed to effectively remove the SCC */

FIG. 1

- ❖ An example for a custom replaceable component class 200 implemented for presenting pie chart SCC for an investment portfolio component 209 for a given account number.
- ❖ At line 1, an object instance 207 initialized by passing an account number (e.g. 40021729) for building SCC the portfolio pie chart 209 for the given investment account. Line 2 adds the SCC to a parent to present the SCC.

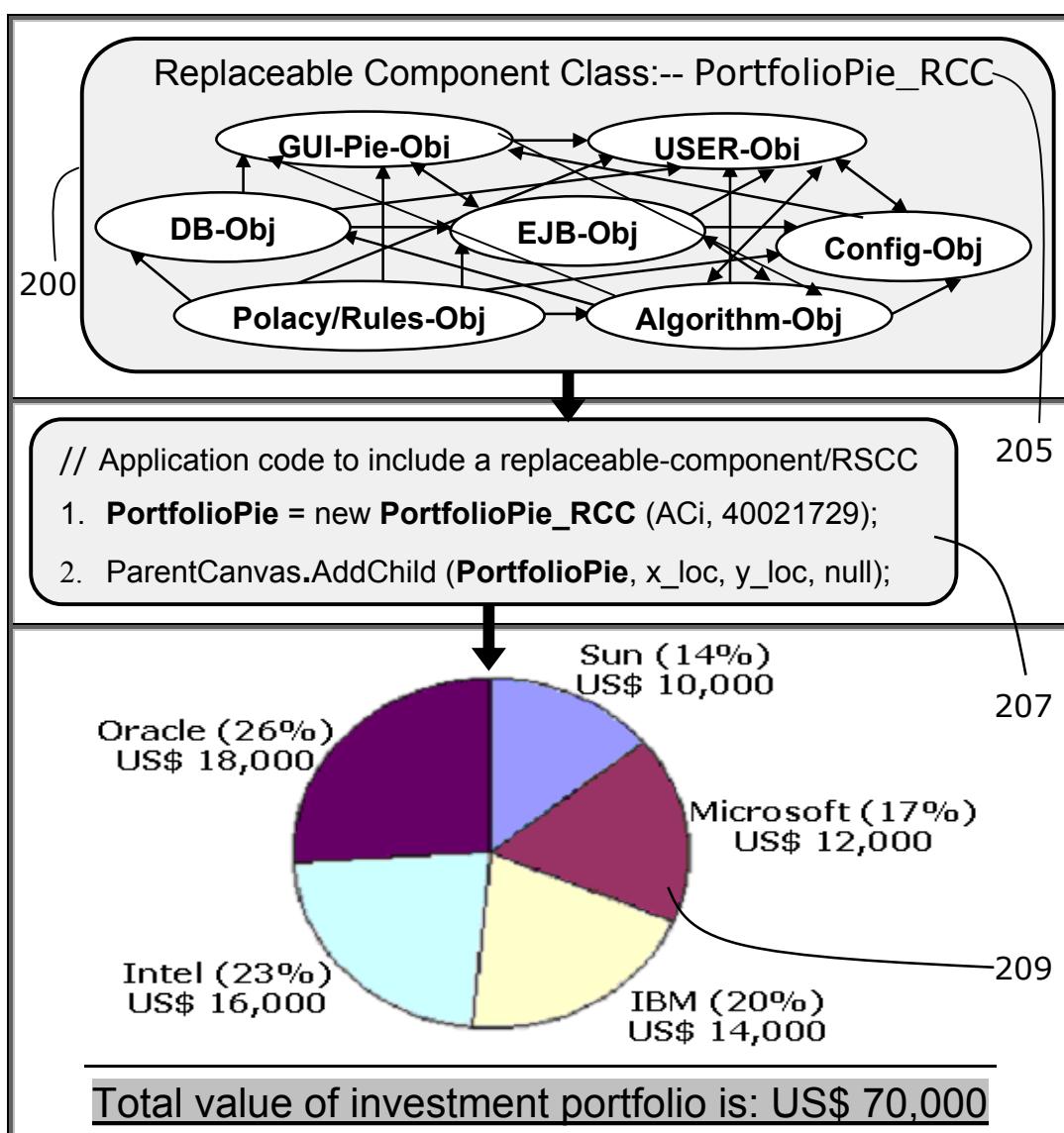


FIG. 2

Using reusable class for GUI-components to build a Container-Component

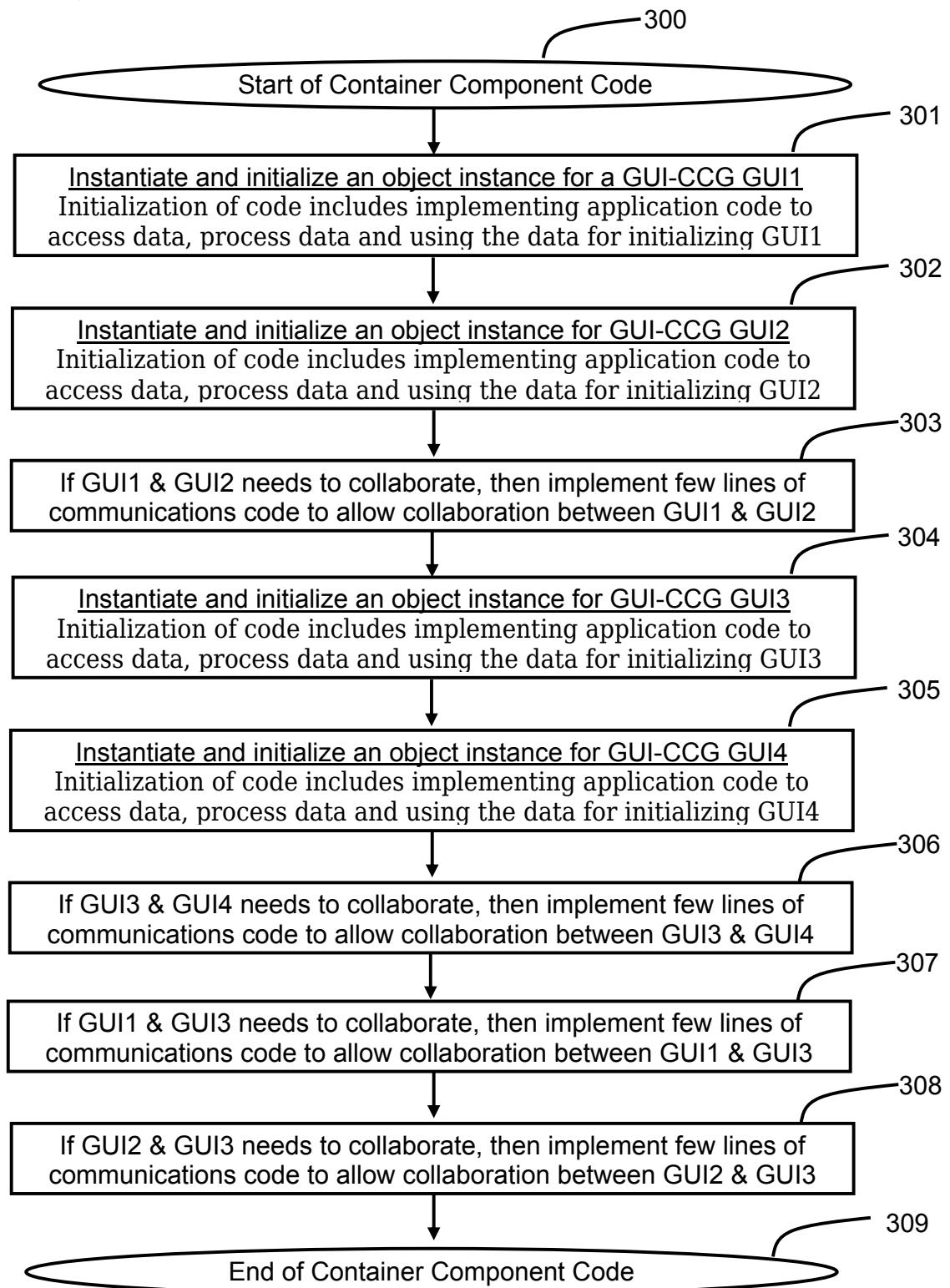
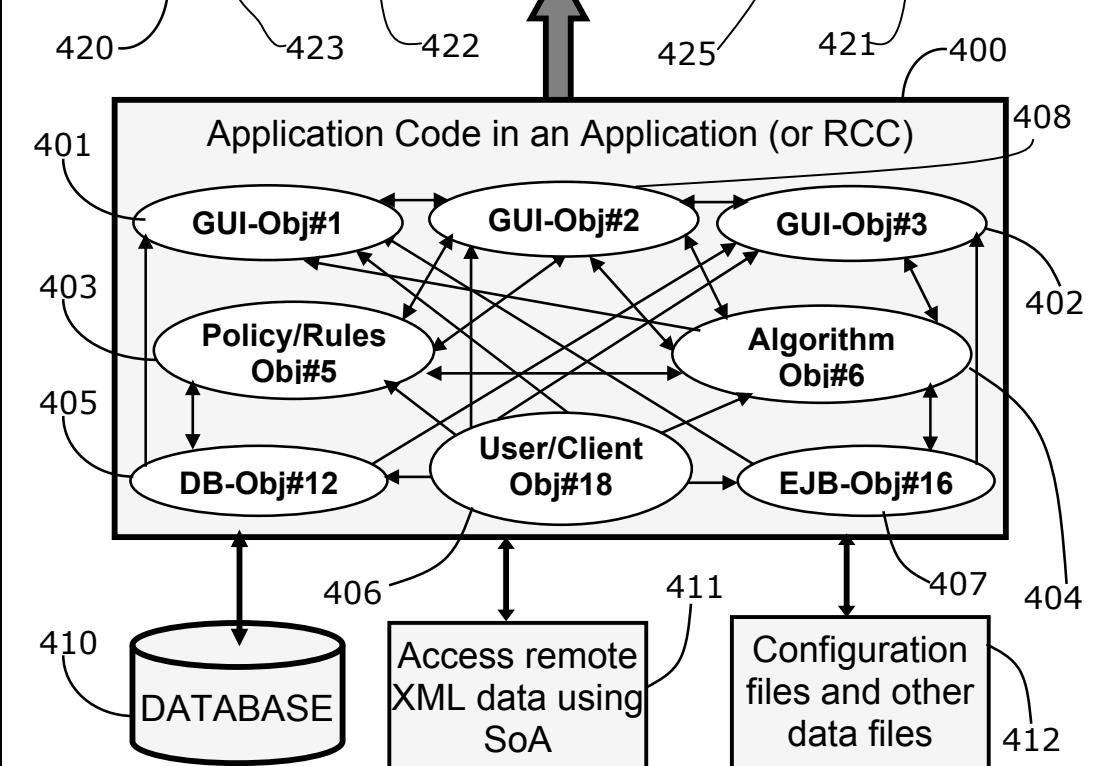
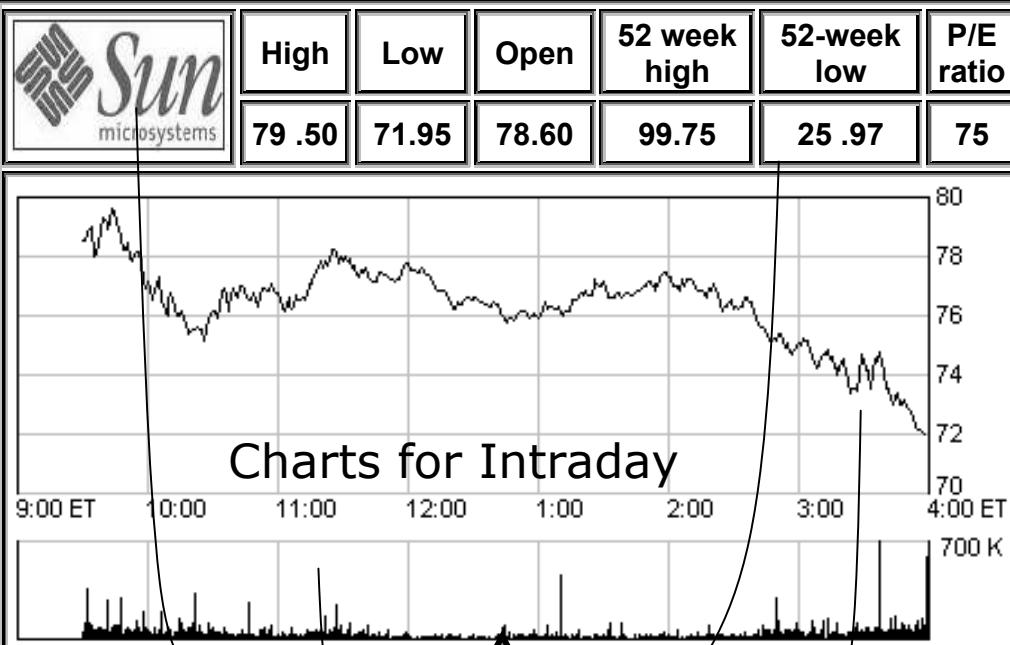


FIG. 3

Below is SCC for presenting quote for Ticker Symbol "SUNW"



Above is RCC to present quote SCC for any given Ticker Symbol

FIG. 4

**A RCC of a container-SCC that uses 3-RCCs for including 3-sub-SCC
(The pseudo-code assuming that the SCC don't need to collaborate with each other)**

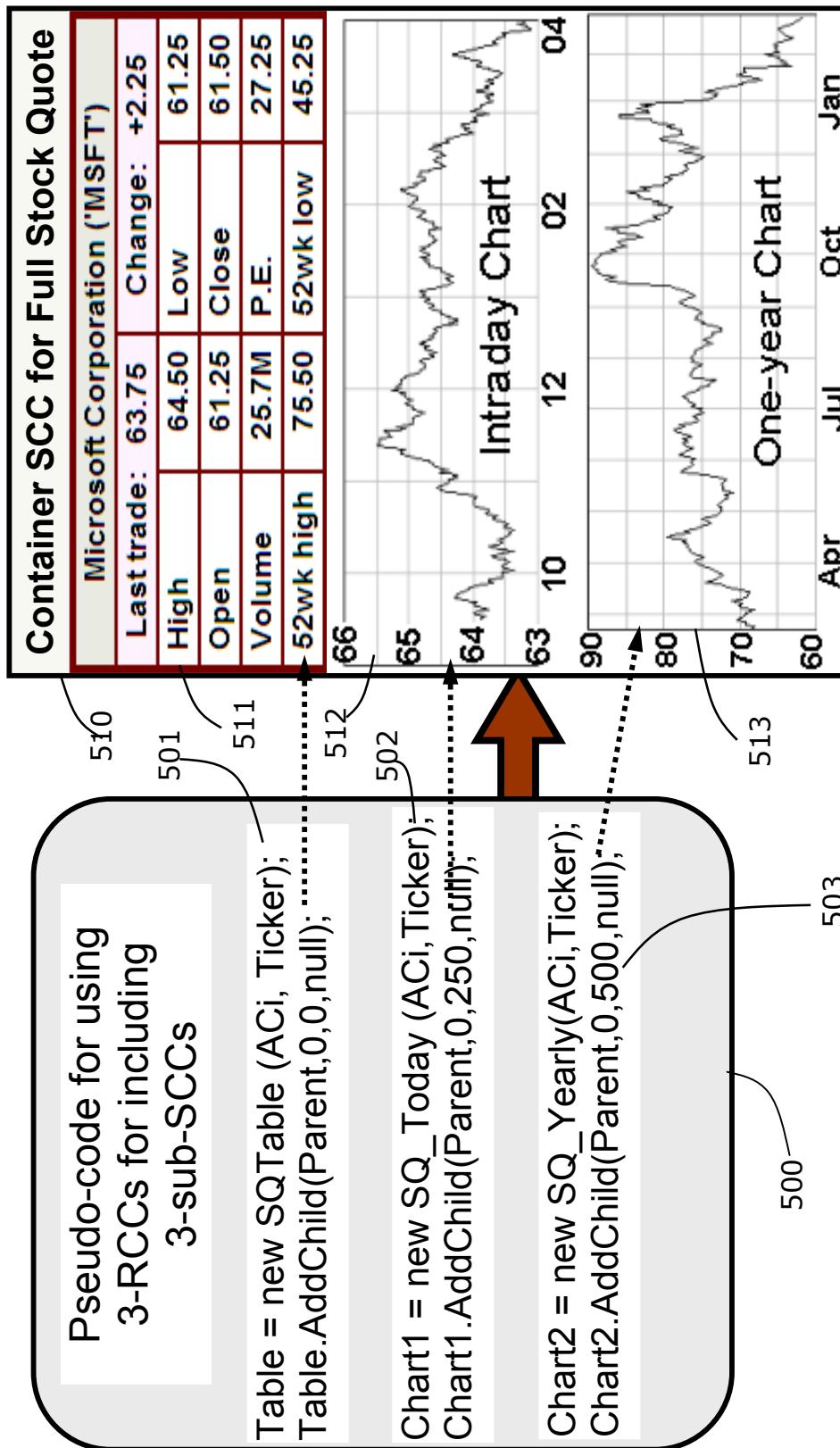


FIG. 5

Building Container-Components by assembling RSCC

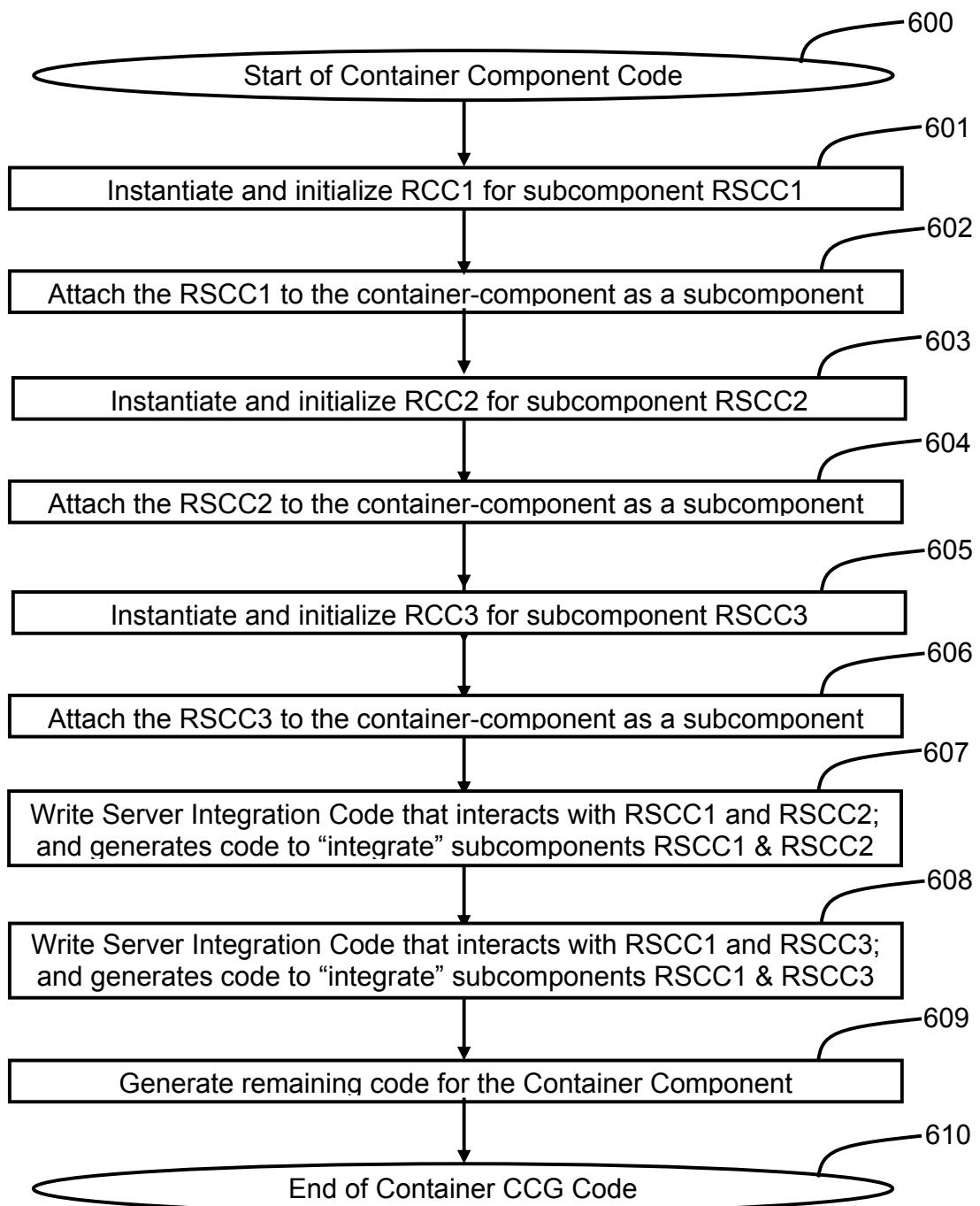


FIG. 6

Construction code of each SCC is encapsulated in a RCC, and an initialized object instance of each RCC is used for including each of the SCCs

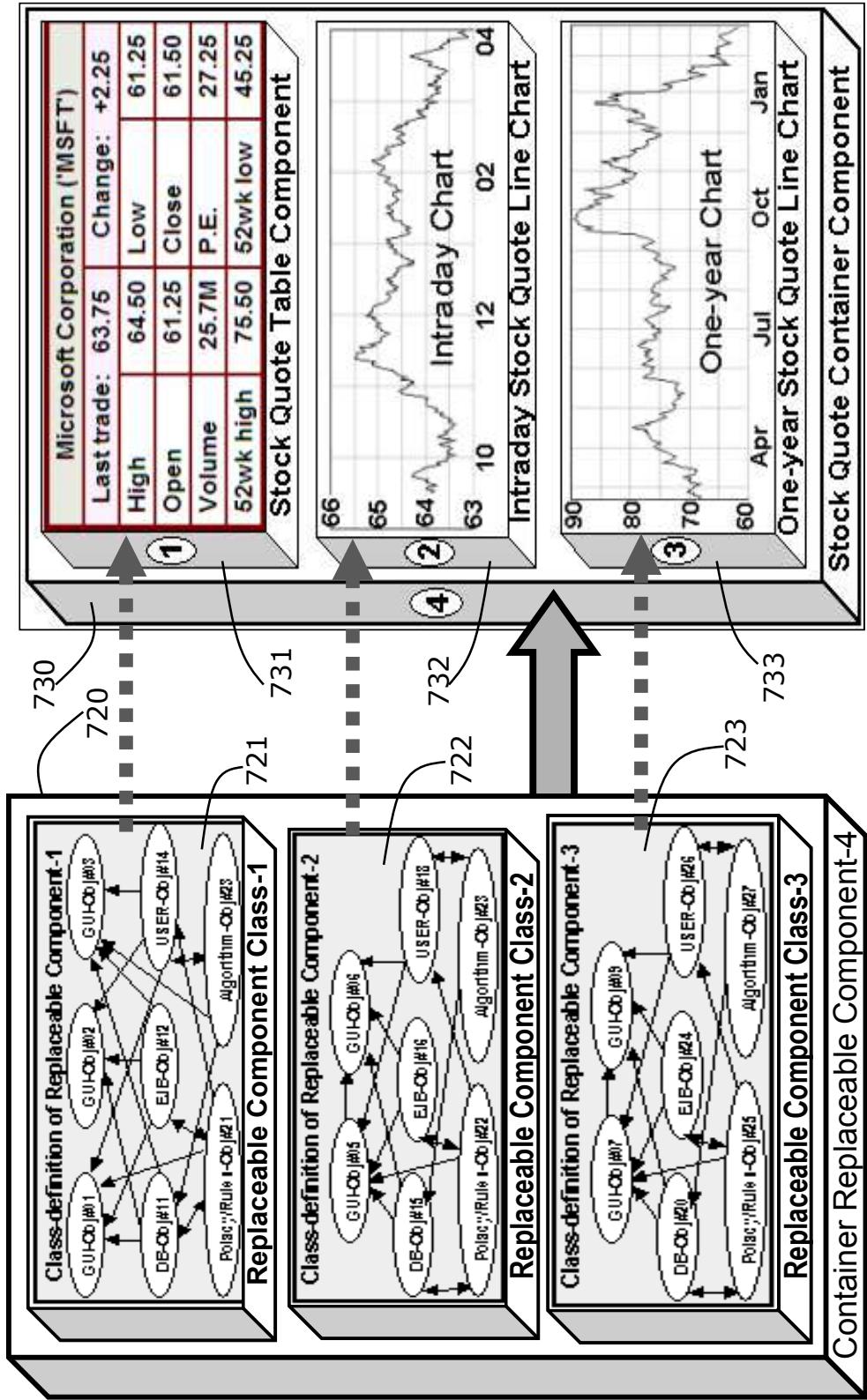


FIG. 7

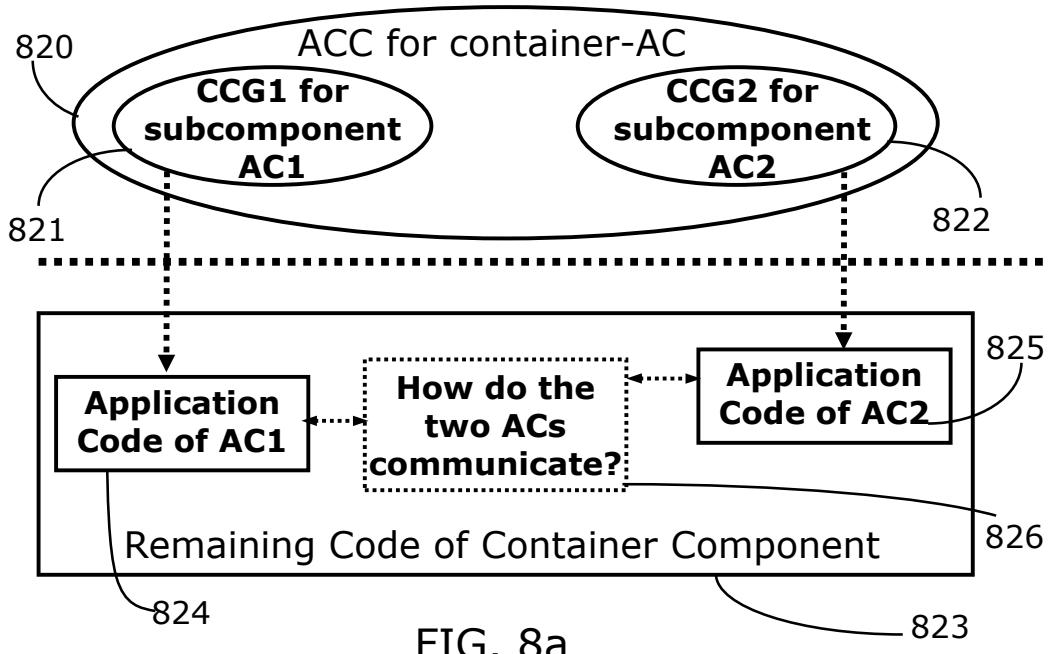


FIG. 8a

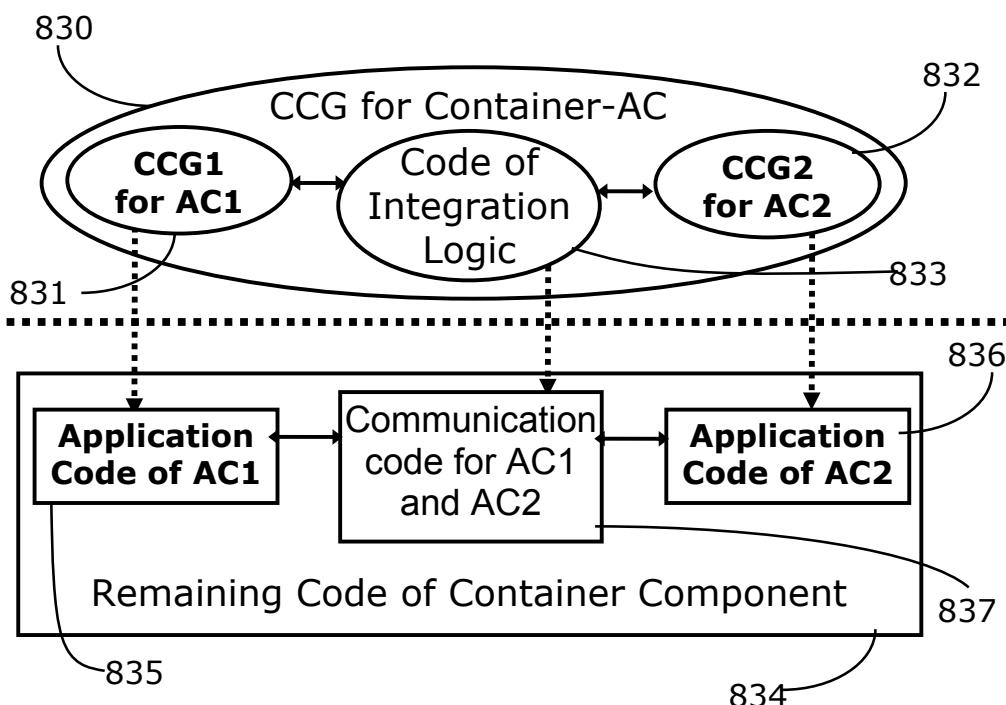


FIG. 8b

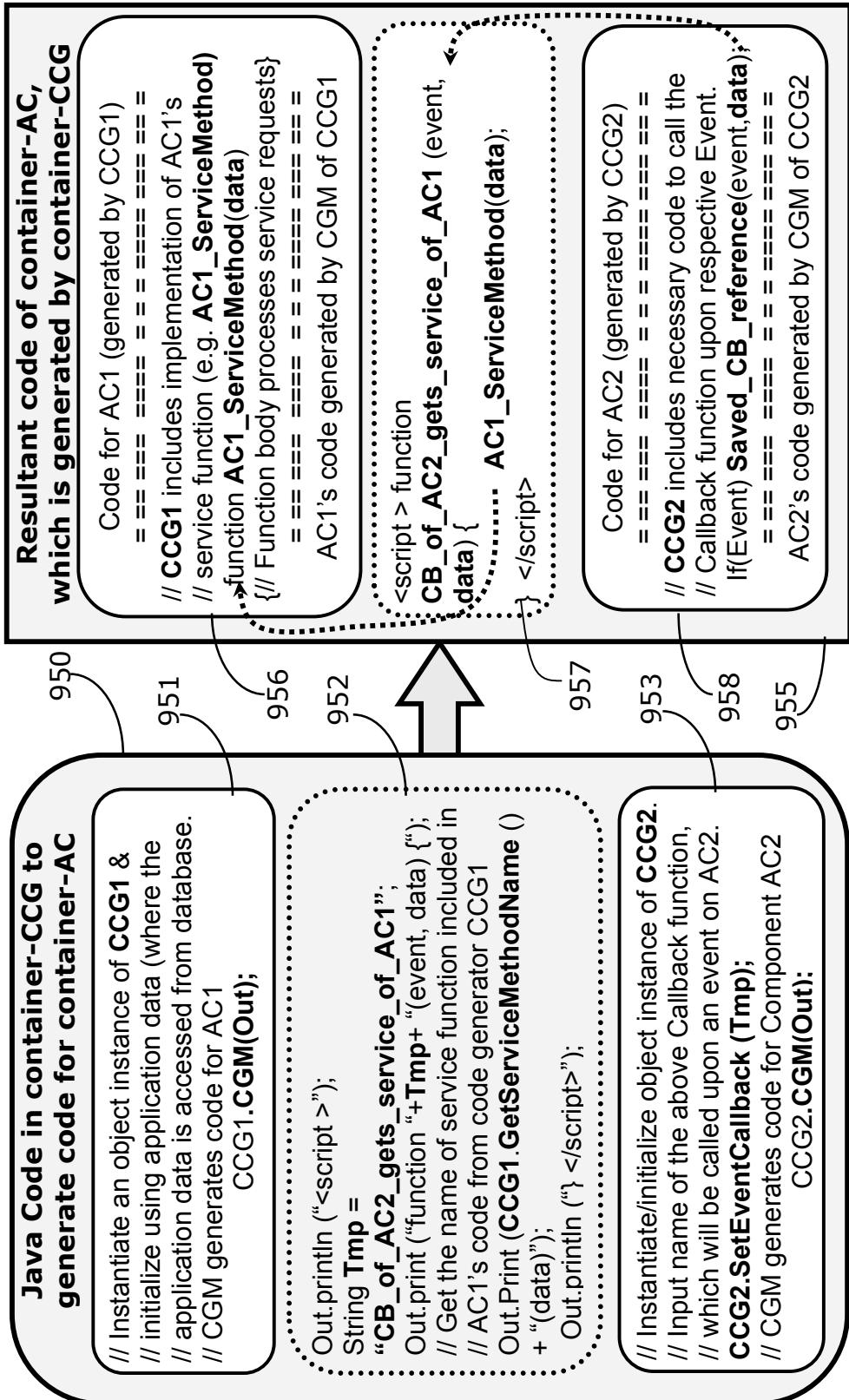


FIG. 9

Server application 1000 using a hierarchy of RCCs (RCC-01 to RCC-20) for generating client GUI application 1040 containing hierarchy of SCCs (AC-01 to AC-20), where AC-01 is generated by RCC-01; AC-02 is generated by RCC-02; AC-03 is generated by RCC-03; and so on

