What is claimed is:

1. A method for providing a container application for a servicing platform, the method being implemented by at least one processor, the method comprising:

displaying, by the at least one processor at each of a plurality of workstations that correspond to users of a container application for use in a contact center environment, a user interface;

receiving, by the at least one processor from a first workstation of a first user via the user interface, a first software application configured to be integrated with the container application for use in the contact center environment;

receiving, by the at least one processor from a second workstation of a second user via the user interface, a second software application configured to be integrated with the container application for use in the contact center environment, wherein the second software application is received contemporaneously with the first software application;

receiving, by the at least one processor from a third user via the user interface, a request to access at least one of the first software application provided by the first user and the second software application provided by the second user;

configuring, by the at least one processor, the container application to allow access to the first software application and the second software application via the container application, wherein the container application allows access to a plurality of software applications;

routing, by the at least one processor, the third user to the container application to access one of the first software application or the second software application;

facilitating, by the at least one processor, a use of each of the first software application and the second software application by each of the first user, the second user, and the third user via the container application; and

displaying, by the at least one processor, to the third user one or more microfront ends for the one of the first software application or the second software application,

wherein only microfront ends where the third user has allowed roles are displayed, and

wherein the container application is a single application and includes each of a plurality of container capabilities including: (1) accommodation of any framework as web components, (2) provision of OAUTH2 authentication and authorization, (3) framework agnostic software development kit (SDK) for widgets, (4) lifecycle events, (5) seamless high availability (HA)/resiliency, (6) seamless login across different seal identifications, (7) dynamically loads layout based on a role, (8) dynamically loads the widgets, (9) provides inter-component eventing, (10) websocket eventing, (11) access to Praesto interactive servicing fabric (ISF) capabilities, and (12) application ID for each microfront end and roles for each layout and applications.

2. The method of claim 1, wherein the displaying of the user interface comprises displaying a microfront end that implements a plurality of microservices that correspond to a subset of the plurality of software applications.

3. The method of claim 1, wherein the facilitating comprises providing, for each of the first software application and the second software application, an access authentication to each of the first user, the second user, and the third user.

4. The method of claim 1, wherein the facilitating comprises providing a communication context that enables a flow of communication between the first software application and the second software application.

5. The method of claim 4, further comprising using a scripting language application programming interface (API) and scripting language events to facilitate the flow of communication.

6. The method of claim 1, wherein the container application includes a container software development kit (SDK) that provides access to a scripting language library and facilitates interactions among the plurality of software applications.

7. The method of claim 1, wherein the plurality of software applications includes software applications that includes a scripting framework.

8. The method of claim 1, further comprising:

when a customer call is received, routing, via the container application, information that relates to the received customer call to each of the plurality of software applications.

9. A computing apparatus for providing a container application for a servicing platform, the computing apparatus comprising:

a processor;

a memory stores instructions; and

a communication interface coupled to each of the processor and the memory,

wherein the processor executes the instructions to perform:

displaying, at each of a plurality of workstations that correspond to users of a container application for use in a contact center environment, a user interface;

receiving, from a first workstation of a first user via the user interface and the communication interface, a first software application configured to be integrated with the container application for use in the contact center environment;

receiving, from a second workstation of a second user via the user interface and the communication interface, a second software application configured to be integrated with the container application for use in the contact center environment, wherein the second software application is received contemporaneously with the first software application;

receiving, by the at least one processor from a third user via the user interface, a request to access at least one of the first software application provided by the first user and the second software application provided by the second user;

configuring, by the at least one processor, the container application to allow access to the first software application and the second software application via the container application, wherein the container application allows access to a plurality of software applications;

routing, by the at least one processor, the third user to the container application to access one of the first software application or the second software application;

facilitating a use of each of the first software application and the second software application by each of the first user, the second user, and the third user via the container application; and

displaying, by the at least one processor, to the third user one or more microfront ends for the one of the first software application or the second software application,

wherein only microfront ends where the third user has allowed roles are displayed, and

wherein the container application is a single application and includes each of a plurality of container capabilities including: (1) accommodation of any framework as web components, (2) provision of OAUTH2 authentication and authorization, (3) framework agnostic software development kit (SDK) for widgets, (4) lifecycle events, (5) seamless high availability (HA)/resiliency, (6) seamless login across different seal identifications, (7) dynamically loads layout based on a role, (8) dynamically loads the widgets, (9) provides inter-component eventing, (10) websocket eventing, (11) access to Praesto interactive servicing fabric (ISF) capabilities, and (12) application ID for each microfront end and roles for each layout and applications.

10. The computing apparatus of claim 9, wherein the processor further performs displaying a microfront end that implements a plurality of microservices that correspond to a subset of the plurality of software applications.

11. The computing apparatus of claim 9, wherein the processor further performs providing, for each of the first software application and the second software application, an access authentication to each of the first user, the second user, and the third user.

12. The computing apparatus of claim 9, wherein the processor further performs providing a communication context that enables a flow of communication between the first software application and the second software application.

13. The computing apparatus of claim 12, wherein the processor further performs using a scripting language application programming interface (API) and scripting language events to facilitate the flow of communication.

14. The computing apparatus of claim 9, wherein the container application includes a container software development kit (SDK) that is configured to provide access to a scripting language library and to facilitate interactions among the plurality of software applications.

15. The computing apparatus of claim 9, wherein the plurality of software applications includes software applications that includes a scripting framework.

16. The computing apparatus of claim 9, wherein when a customer call is received, the processor further performs routing, via the container application, information that relates to the received customer call to each of the plurality of software applications.

17. A non-transitory computer readable storage medium storing instructions for providing a container application for a servicing platform, the storage medium comprising executable code which, when executed by a processor, causes the processor to:

display, at each of a plurality of workstations that correspond to users of a container application for use in a contact center environment, a user interface;

receive, from a first workstation of a first user via the user interface, a first software application configured to be integrated with the container application for use in the contact center environment;

receive, from a second workstation of a second user via the user interface, a second software application configured to be integrated with the container application for use in the contact center environment, wherein the second software application is received contemporaneously with the first software application;

receive, from a third user via the user interface, a request to access at least one of the first software application provided by the first user and the second software application provided by the second user;

configure the container application to allow access to the first software application and the second software application via the container application, wherein the container application allows access to a plurality of software applications;

route the third user to the container application to access one of the first software application or the second software application;

facilitate a use of each of the first software application and the second software application by each of the first user, the second user, and the third user via the container application; and

display, to the third user, one or more microfront ends for the one of the first software application or the second software application,

wherein only microfront ends where the third user has allowed roles are displayed, and

wherein the container application is a single application and includes each of a plurality of container capabilities including: (1) accommodation of any framework as web components, (2) provision of OAUTH2 authentication and authorization, (3) framework agnostic software development kit (SDK) for widgets, (4) lifecycle events, (5) seamless high availability (HA)/resiliency, (6) seamless login across different seal identifications, (7) dynamically loads layout based on a role, (8) dynamically loads the widgets, (9) provides inter-

component eventing, (10) websocket eventing, (11) access to Praesto interactive servicing fabric (ISF) capabilities, and (12) application ID for each microfront end and roles for each layout and applications.

18. The storage medium of claim 17, wherein the executable code is further configured to cause the processor to display a microfront end that implements a plurality of microservices that correspond to a subset of the plurality of software applications.

19. The storage medium of claim 17, wherein the executable code is further configured to cause the processor to provide, for each of the first software application and the second software application, an access authentication to each of the first user, the second user, and the third user.

20. The storage medium of claim 17, wherein the container application includes a container software development kit (SDK) that provides access to a scripting language library and facilitates interactions among the plurality of software applications.