

We claim

1. A non-transitory computer readable storage medium containing executable program code by a processor to process requests of an application service provider: receive, by a mother script, a request from an input device; at least by the processor, generate, spawn, or invoke by the mother script daughter script containing adaptive instruction sets in which different system-level requests arrange data and corresponding instructions differently, determining, based on a parameter in a system data, whether the daughter script should be executed in a virtual machine; creating the virtual machine for the daughter script based on the parameter; wherein the daughter script is configured to generate additional scripts as the processing of the daughter script continues; determining, by the additional script, a device to receive a user interface based on the received request providing, by the additional scripts, the user interface to the determined device; and wherein the daughter script virtual machine are destroyed by themselves.
2. The computer readable storage medium of claim 1, wherein the mother script includes at least one adaptive instruction set that dynamically generates, spawns, or invokes, one or more daughter or virtual scripts based at least in part on data collected by the mother script, data stored in a system database, or data stored in a user's database.
3. The computer readable storage medium of claim 1 including programming to provide at least one template configured to be populated by one of a messaging and collaboration component and a database management component.

4. The computer readable storage medium of claim 1, further comprising programming to provide a dynamic user interface.
5. The computer readable storage medium of claim 1, further comprising programming to provide a dynamic encryption module that is configured to change periodically or randomly.
6. The computer readable storage medium of claim 1, further comprising programming to provide a dynamic decryption module in its own virtual machine separate and distinct from a main database and executed in an operating system environment that is completely separate from an environment used to execute any other scripts.
7. The computer readable storage medium of claim 1, further comprising programming to provide decrypt and validate data in its own virtual machine separate and distinct from a main database and executed in an operating system environment that is completely separate from an environment used to execute any other scripts.
8. The computer readable storage medium of claim 1, further comprising programming to track the transmission of data with timestamps.
9. The computer readable storage medium of claim 1, wherein a user's characteristics are maintained in a historical database including a record for each user in a cloud environment.

10. The computer readable storage medium of claim 1, wherein data is stored in a storage area that is allocated to an Application Service Provider by a cloud computing provider.

11. The computer readable storage medium of claim 1, further comprising programming to dynamically push or spawn a decryption agent on demand as a daughter web script via a virtual machine to the user when called by a request from a voice-based or data-driven device.

12. The computer readable storage medium of claim 1, further comprising programming to store decrypted data on a virtual disk in a virtual machine or as a file on a disk on a physical server.

13. The computer readable storage medium of claim 1, further comprising programming to dynamically query a storage of data or data store remotely in real time for presentation on a destination device.

14. The computer readable storage medium of claim 1, further comprising programming to provide advertising automatically in a selected language or demographic.

15. The computer readable storage medium of claim 1, further comprising programming to provide a dynamic user interface in a first language via first input format, capture data using the first input formation, translate the data into a second language, and provide the data to a user in the first, second, or a third language.

16. A method of processing requests of an application service provider, comprising: receiving by a mother script, a request from an input device; at least by the processor, generating, spawning, or invoking by the mother script a daughter script containing adaptive instruction sets in which different system-level requests arrange data and corresponding instructions differently, determining, based on a parameter in a system data, whether the daughter script should be executed in a virtual machine; creating the virtual machine for the daughter script based on the parameter; generating, from the daughter script additional scripts as the processing of the daughter script continues; determining, by the additional script, a device to receive a user interface based on the received request; providing, by the additional scripts, the user interface to the determined device; and destroying by the processor, the daughter script virtual machines by themselves.

17. The method of claim 16, wherein the mother script includes at least one adaptive instruction set that dynamically generates, spawns, or invokes, one or more daughter or virtual scripts based at least in part on data collected by the mother script, data stored in a system database, or data stored in a user's database.

18. The method of claim 16, further comprising populating at least one template by one of a messaging and collaboration component and a database management component.

19. The method of claim 16, further comprising providing a dynamic user interface.

20. The method of claim 16, further comprising providing a dynamic encryption module that is configured to change periodically or randomly.