We claim :

1. A method of enforcing policy limitations on an execution of an application program within a first computing environment, the method comprising:

obtaining application policy data, corresponding to the application program, from a second computing environment differing from the first computing environment, the application policy data specifying a first precondition that is to be bypassed when the first computing environment is a container virtual computing environment;

performing environment type validation comprising at least one of:

checking an encryption infrastructure of the first computing environment;

checking a code integrity infrastructure of the first computing environment; or

checking for hypervisor functionality within the first computing environment; and

limiting the execution of the application program based on the first precondition when the performed environment type validation determines that the first computing environment is not a container virtual computing environment.

2. The method of claim 1, wherein the checking the encryption infrastructure comprises verifying a proper attestation of an operating system of the first computing environment.

3. The method of claim 2, wherein the checking the encryption infrastructure comprises verifying a proper attestation of a combination of the operating system of the first computing environment and the application program.

4. The method of claim 2, wherein the verifying the proper attestation of the operating system of the first computing environment comprises communicating with a remote attestation service that is external to the first computing environment.

5. The method of claim 2, wherein the verifying the proper attestation of the operating system of the first computing environment comprises comparing attestation measurements from a first Trusted Platform Module (TPM) in the first computing environment to attestation measurements from a second TPM in the second computing environment.

6. The method of claim 1, wherein the checking the encryption infrastructure comprises verifying that a certificate in a first certificate store of the first computing environment is signed by a pre-determined root of trust.

7. The method of claim 1, wherein the checking the code integrity

infrastructure comprises verifying a hash of an active code integrity policy file against a security catalog comprising hashes of known good code integrity policy files.

8. The method of claim 7, wherein the checking the code integrity infrastructure comprises verifying that the security catalog is signed by one of a predetermined trusted organization's certificate.

9. The method of claim 1, wherein the checking for the hypervisor functionality comprises checking whether processor functionality commensurate with a hypervisor is exposed by a central processing unit of the first computing environment.

10. The method of claim 1, wherein the performing the environment type validation comprises checking whether a predetermined system driver component is present within the first computing environment.

11. The method of claim 1, wherein the performing the environment type validation comprises checking whether a resource management stack is deactivated within the first computing environment.

12. The method of claim 1, wherein the performing the environment type validation comprises checking whether a predetermined firmware variable is set to a predetermined value within a firmware of the first computing environment.

13. The method of claim 1, wherein the first precondition limits execution of the application program to a specific computing device hardware machine identifier.

14. The method of claim 1, wherein the first precondition limits execution of the application program to a specific user identifier.

15. The method of claim 1, further comprising verifying the obtained application policy data by reference to a policy management service external to the first computing environment.

16. A computing device comprising :

one or more processing units; and

one or more non-transitory computer-readable storage media comprising computer-executable instructions, which, when executed by at least some of the processing units, cause the computing device to :

obtain application policy data, corresponding to the application program, from a second computing environment differing from the first computing environment, the application policy data specifying a first precondition that is to be bypassed when the first computing environment is a container virtual computing environment;

perform environment type validation comprising at least one of :

checking an encryption infrastructure of the first computing environment;

checking a code integrity infrastructure of the first computing environment; or

checking for hypervisor functionality within the first computing environment; and

limit the execution of the application program based on the first precondition when the performed environment type validation determines that the first computing environment is not a container virtual computing environment.

17. The computing device of claim 16, wherein both the first computing environment and the second computing environment are hosted by the computing device.

18. The computing device of claim 16, wherein the first computing environment is are hosted by the computing device and the second computing environment is hosted by a second computing device, differing from the first computing device, the second computing device being communicationally coupled to the computing device.

19. One or more non-transitory computer-readable storage media comprising computer-executable instructions, which, when executed within a first computing environment, perform steps comprising :

receiving a request to validate a type of the first computing environment;

performing at least one of :

checking an encryption infrastructure of the first computing environment;

checking a code integrity infrastructure of the first computing environment; or

checking for hypervisor functionality within the first computing environment; and

generating an environment type validation Boolean output value indicative of the first computing environment not being a container virtual computing environment if at least one of the checking of the encryption infrastructure or the checking of the code integrity infrastructure failed, or if the checking for the hypervisor functionality confirmed hypervisor functionality within the first computing environment.

20. The one or more non-transitory computer-readable storage media of claim 19, wherein the computer-executable instructions, when executed, perform further steps comprising at least one of :

checking whether a predetermined system driver component is present within the first computing environment;

checking whether a resource management stack is deactivated within the first computing environment; or

checking whether a predetermined firmware variable is set to a predetermined value within a firmware of the first computing environment.