SYSTEM FOR AND METHOD OF DETERMINING DATA CONNECTIONS BETWEEN SOFTWARE APPLICATIONS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This International Application claims priority of U.S. Application No. 16/051,660 filed August 1, 2018, the disclosure of which is expressly incorporated by reference herein in its entirety.

FIELD

[0002] The present invention relates generally to a system for and a method of determining data connections between software applications as those applications are built rather than after their complete deployment.

BACKGROUND

[0003] Software applications very rarely run in isolation. In fact, software applications, particularly applications that process data from diverse sources, may receive and send data to dozens of different locations throughout a series of connected resources. If the application code is operating as intended, this interaction doesn't generally cause problems. However, should the application fail to operate as intended or if other applications with which the application interacts fail, the result may be data corruption or erroneous results. The impact of corrupt data or erroneous results that are relied on to the detriment of an organization can be costly. Current methods of tracking interactions between applications involve adding intended data connections into program requirements documents and test plans. These amount to intentions rather than results as coding errors or unintended program operation can cause connections to break or perform in an unexpected manner. Thus, existing methods of documenting connections between applications can result in lack of visibility to actual connections, the inability to trace connection issues and the inability to validate interconnections in an application that has been introduced into production. What is needed is a system for and a method of defining application connectivity in such a way that it is self-regulating.

SUMMARY

[0003a] According to a first aspect of the present invention, there is provided a method of controlling data connections of an application program, the method comprising: establishing a service definition for the application program corresponding to an application development phase; establishing definitions of allowed connections; storing

the service definition and definitions of allowed connections in an application service registry; embedding the definitions of allowed connections as metadata in a source code for the application program; automatically deriving firewall rules from the metadata by, identifying a plurality of communication endpoints, the plurality of communication endpoints including the application program; extracting the metadata corresponding to each of the plurality of communication endpoints; and determining whether a connection between each of the plurality of communication endpoints is permitted based on a comparison of the extracted metadata; automatically deriving an allowed application data listing from the metadata; and configuring an application interface manager using the allowed application data listing.

[0003b] According to a second aspect of the present invention, there is provided a system for automatically regulating data connections of an application program, the system comprising: an application service registry; a source data repository in communication with the application service registry; the application service registry comprising software instructions that when executed by a first processor cause the first processor to: extract metadata information pertaining to permitted application data connections; the source data repository comprising software instructions that when executed by a second processor, cause the second processor to: establish a software application requirements definition for the application program corresponding to an application development phase, the software application requirements definition comprises allowed connections with other applications and data sources; store the definitions of allowed connections to a metadata document; embed the definitions of allowed connections as metadata into a source code for the application program; automatically establish firewall rules from the metadata to, identify a plurality of communication endpoints, the plurality of communication endpoints including the application program; extract the metadata corresponding to each of the plurality of communication endpoints; and determine whether a connection between each of the plurality of communication endpoints is permitted based on a comparison of the extracted metadata; automatically establish an allowed application data listing from the metadata; and configure an application interface manager using the allowed application data listing.

[0004] In exemplary embodiments, connectivity between an application and other applications or data sources is encoded in the source code (source code, object code, scripts, interpreted code, and the like) such that these relationships are defined and can be used to automatically generate connectivity maps. This allows information flow to be reliably

CLAIMS

What is claimed is:

1. A method of controlling data connections of an application program, the method comprising:

establishing a service definition for the application program corresponding to an application development phase;

establishing definitions of allowed connections;

storing the service definition and definitions of allowed connections in an application service registry;

embedding the definitions of allowed connections as metadata into a source code for the application program;

automatically deriving firewall rules from the metadata by,

identifying a plurality of communication endpoints, the plurality of communication endpoints including the application program;

extracting the metadata corresponding to each of the plurality of communication endpoints; and

determining whether a connection between each of the plurality of communication endpoints is permitted based on a comparison of the extracted metadata;

automatically deriving an allowed application data listing from the metadata; and

configuring an application interface manager using the allowed application data listing.

2. The method of claim 1, wherein the definitions of allowed connections are also embedded as metadata in a corresponding application program.

3. The method of claim 2, further comprising creating a definition of data publications and data sources and wherein a databook is created that comprises definitions of critical data elements and data sourcing rules derived from the definition of data publications and data sources.

4. The method of claim 1, wherein the step of establishing definitions of allowed connections comprises identifying application types to which an application program is permitted to communicate.

5. The method of claim 1, wherein the step of establishing definitions of allowed connections comprises identifying application environments to which the application program is permitted to communicate.

6. The method of claim 1, wherein the step of establishing definitions of allowed connections comprises identifying ports through which the application program is permitted to communicate.

7. The method of claim 1, wherein the step of establishing definitions of allowed connections comprises defining geographic locations with which the application program is permitted to communicate.

8. The method of claim 1, wherein the step of establishing definitions of allowed connections comprises defining at least one data type from among an environmental variable, a location variable, a confidentiality variable, and a regulatory and compliance variable that may be communicated from the application program.

9. The method of claim 1, further comprising the steps of:

comparing an attempt to access a connection by an application program running in a test environment with the definitions of allowed connections; and

recording each attempt to access a connection that is not included in the definition of allowed connections.

10. A system for automatically regulating data connections of an application program, the system comprising:

an application service registry;

a source data repository in communication with the application service registry;

the application service registry comprising software instructions that when executed by a first processor cause the first processor to:

extract metadata information pertaining to permitted application data connections;

the source data repository comprising software instructions that when executed by a second processor, cause the second processor to:

establish a software application requirements definition for the application program corresponding to an application development phase, the software application requirements definition comprises allowed connections with other applications and data sources;

store the definitions of allowed connections to a metadata document;

embed the definitions of allowed connections as metadata into a source code for the application program;

automatically establish firewall rules from the metadata to,

identify a plurality of communication endpoints, the plurality of communication endpoints including the application program;

extract the metadata corresponding to each of the plurality of communication endpoints; and

determine whether a connection between each of the plurality of communication endpoints is permitted based on a comparison of the extracted metadata;

automatically establish an allowed application data listing from the metadata; and

configure an application interface manager using the allowed application data listing.

11. The system of claim 10, further comprising software instructions that cause the second processor to create a definition of data publications and data sources.

12. The system of claim 10, further comprising software instructions that cause the second processor to create a databook that comprises definitions of critical data elements and data sourcing rules derived from the definition of data publications and data sources.

13. The system of claim 10, wherein the step of establishing a software application requirements definition further comprises identifying allowed connections with other applications and data sources.

14. The system of claim 10, wherein the step of establishing a software application requirements definition further comprises identifying application environments to which the application program is permitted to communicate.

15. The system of claim 10, wherein the step of establishing a software application requirements definition further comprises defining ports through which the application program is permitted to communicate.

16. The system of claim 10, wherein the step of establishing a software application requirements definition further comprises defining geographic locations from which the application program is permitted to communicate.

17. The system of claim 10, wherein the step of establishing a software application requirements definition further comprises defining at least one data type from among an environmental variable, a location variable, a confidentiality variable, and a regulatory and compliance variable that may be communicated from the application program.

18. The system of claim 10, further comprising an application interface manager, the application interface manager comprising software instructions that when executed by the application interface manager cause the application interface manager to:

compare an attempt to access a connection by an application program running in a test environment with the definitions of allowed connections; and

record access attempts to connections that are not including in the definition of allowed connections.