

the privilege or access allowed for the application program **104** for Resource A **106**. In this example, a memory area stores the manifest **108**. The manifest **108** maps an application identifier and a resource to a privilege. The manifest **108** stores the privilege (e.g., privilege X) as a function of the application identifier (e.g., **ID1**) and the resource (e.g., Resource A **106**). Privilege X may correspond, for example, to read-only access. The operating system **102** provides the application identifier with access to Resource A **106** according to the determined privilege.

[**0027**] In one embodiment, the operating system **102** stores or has access to one or more computer-executable components on a computer-readable media. A processor associated with the operating system **102** is configured to execute the computer-executable components or other computer-executable instructions to determine, responsive to a request from the application program **104** for the resource, the privilege from the manifest **108** stored in the memory area as a function of the application identifier and the resource. The processor is further configured to execute computer-executable instructions to grant the application program **104** access to the resource or a copy thereof according to the determined privilege.

[**0028**] In particular, the computer-executable components grant the application program **104** with access to the resource. In the particular embodiment of **FIG. 1**, the components include an interface module **110**, an identity module **112**, a filter module **114**, and an access control module **116**. The modules in **FIG. 1** may exist separate from and independent of the operating system **102**. Further, the functionality and structure of embodiments of the invention may be organized into any quantity of modules, components, or the like. For example, the modules may be distributed.

[**0029**] The interface module **110** receives a request from the application program **104** for access to the resource identified in the request. In one embodiment, the interface module **110** receives the request from the application program **104** for access to one or more of the following: a file, a directory, and a system setting (e.g., a registry entry). The identity module **112** determines an application identifier for the application program **104** to distinguish the application program **104** and components thereof from other application programs. In one embodiment, the identity module **112** determines the application identifier (e.g., an isolation identifier) for a group of application programs. As the application program **104** may include a plurality of files and system settings, the identity module **112** determines the application identifier associated with each of the plurality of files and system settings representing the application program **104**. The filter module **114** identifies a privilege from the manifest **108** as a function of the application identifier determined by the identity module **112** and the identified resource. The manifest **108** indicates the privilege that the application program **104** has for accessing the identified resource. The access control module **116** grants the application program **104** access to the identified resource according to the privilege identified by the filter module **114**. In one embodiment, a configuration module receives an application manifest from an installation medium associated with the application program **104**. The application manifest represents a list of files and resource changes (e.g., system settings) associated with the application program **104**. The configuration module

may update an operating system manifest with the data contained in the application manifest. Alternatively, the configuration module may maintain each application manifest for each installed application.

[**0030**] Manifest

[**0031**] The manifest **108** includes a list of items (e.g., files and resource changes) or objects associated with an application program **104** or an operating system such as operating system **102**. Alternatively, the list of items associated with application program **104** may be stored in a configuration file or store for each resource provider. In another embodiment, the creator of the object specifies the access privileges directly on the object.

[**0032**] The manifest **108** may also include a list of privileges for resources associated with the application program **104** or operating system **102**. For example, an author of the application program **104** may specify in the manifest the privileges to resources of the operating system **102** and/or to resources that the application program **104** may create. Alternatively, the manifest may simply store the identity information associated with the application program **104**. In another example, an installation medium storing the application program **104** to be installed may also store an application manifest listing items associated with the application program **104** and privileges associated with application private resources. Third party application vendors or personnel responsible for deployment of the application program **104** may create the application manifest. In another example, an operating system manifest stores a list of items associated with the application programs installed with the operating system **102**. The operating system manifest may further store a list of components associated with the operating system **102**. In one embodiment, the operating system manifest represents the aggregation of the protection behaviors for each of the operating system components or installed application programs. The aggregated manifest defines the types of interaction that will be permitted for each file, directory, and system setting.

[**0033**] The operating system **102** is self-describing in that it specifies how it wants to be protected and how operating system components and other components may interact and extend the system. In one embodiment of the invention, it is possible to declare the type of protection behavior that should be enforced by the operating system **102** for every item or resource (e.g., file, directory, registry key and value, driver, etc.) that is part of the operating system **102**.

[**0034**] The manifest **108** is stored as a data structure on a computer-readable medium. The manifest **108** specifies access rights of application programs such as application program **104** to access a plurality of resources. The exemplary data structure in **FIG. 1** includes a first field storing a value (e.g., **ID1**) representing an identity corresponding to the application program **104**. For example, the first field may store a value based on one or more of the following: a version, a central processing unit, and a public key. The data structure also includes a second field storing a list of resources (e.g., Resource A **106**) associated with the application program **104**. For example, the second field may store a list of resources such as the following: a file, a directory, and a system setting. The data structure also includes a third field storing a privilege (e.g., Privilege X) or other declaration of intent associated with the identity from the first field