11.4 Specific Authorization Requirement Pattern

Basic Details

Related patterns:	Extends user authorization; refers to configurable authorization
Anticipated frequency:	Up to a dozen requirements
Pattern classifications:	None

Applicability

Use the specific authorization requirement pattern to specify that a set of users is authorized (or is not authorized) to do or see certain things.

Do not use the specific authorization requirement pattern to specify that user authorization is to be configurable; use the configurable authorization requirement pattern for that.

Discussion

A specific authorization requirement makes a statement—of any kind—that affects what a certain set of users can see and do. These requirements can vary considerably, but they generally fall into one of the following categories (or possibly more than one—which is most frequently seen in combinations of Categories 2 and 4):

Category 1: Universal denial-by-default rule To assert that nothing of value in the system is accessible unless permission is expressly granted.

Category 2: Functions Functions that certain kinds of people are authorized to use. A requirement can either name an individual function or a class of functions.

Category 3: Actions within functions For example, a user might be authorized to change details but not add or delete something or other.

Category 4: Data To restrict access to certain information.

Category 5: Limits Limits on values a user is allowed to work with—such as a monetary amount up to which they are authorized to perform a certain action (for example, a refund to a customer).

Category 6: Time To restrict *when* certain privileges can be used (what times of day and/or days of the week).

Category 7: Environment To apply restrictions to certain environments (such as remote access).

Category 8: Strength of authentication To allow a user to do more if they logged in using a strong authentication mechanism (such as a smartcard) than they could do if they used a weaker mechanism (such as when they have mislaid the card and just entered their password).

Category 9: Transference Where a restriction applied in one situation is transferred to apply in the same way to something else (such as data copied into a chronicle being subject to the same protection as the original data).

Category 10: Operational rules To achieve such things as division of responsibility (so that the same person can't both perform and then approve a transaction, for instance).

Category 11: Blanket bans Things that *no one* may have access to—mainly functions that are present anyway but which it's bad practice to use day-to-day in a commercial system (such as unrestricted access to the database or operating system).

Category 12: Blanket permission Things that *everyone* can access without restriction. Publicly available things normally don't warrant a separate requirement to say so, but you might encounter a situation that deserves it.

Use this list to help identify what you need: go through each point in turn. Each one can be regarded as an **access rule** that the organization wishes to apply to the operation of its system. (If you'd like to apply certain rules more widely than just within one system, pull them out into a separate requirements specification—"common requirements"—and then refer to them in the requirements for each system to which they apply, as per the refer-to-requirements requirement pattern.) Recognize the bounds of what the system can realistically restrict access to—and don't waste your time trying to control anything beyond those bounds. That might include hardware, other systems, and third-party software used to implement part of the system. You might be forced to make unpleasant compromises.

Content

A specific authorization requirement needs to convey two things: **who** and **what**. The preceding list is of various kinds of *what*. As for *who*, express it in whatever terms are appropriate—but be clear. Don't name particular users (people). Mention roles or job titles, if you like, but if you do, define them either within the requirement or elsewhere (as informal material). This is important, because if you don't, readers are apt to interpret them differently. (Someone else's picture of a "systems administrator" might not accord with yours, for example, aside from theirs having green hair and yours purple.) If you're building a system to be used in multiple places, they might be organized very differently; make it as easy as possible for everyone to see how your system would fit their environment.

A specific authorization requirement should contain:

1. **Privilege description** (*What.*) It can be anything to which access can be granted (or denied), but is most commonly the name of a function.
2. **Access rule** (*Who*, and in which circumstances.) Most commonly, it identifies a **type of user**, and at its simplest, it says that such users are granted the relevant privilege. But put whatever you want into a rule—including conditions and other logic.

Template(s)

Summary	Definition
«*Privilege summary*» access	«*Privilege description*» shall [not] be accessible «*Access rule description*».
«*Privilege summary*» access	A «*Type of user*» shall [not] be able to «*Privilege description*».

Example(s)

The examples here are presented according to the categories listed in this pattern's "Discussion" section (along with a few that are combinations of Categories 2 and 4).

Category 1, a universal denial-by-default rule:

Summary	Definition
Denial of access by default	A user shall have no access to any function or information or other system resource unless they have been explicitly been granted permission, or unless it has been designated publicly accessible. In the case of information designated as publicly accessible, this shall be taken to mean only the ability to *view* the information, unless explicitly specified otherwise.

Category 2, functions (starting with one you might regard as so obvious that it goes without saying, although a stickler could argue that without it, unrestricted access is acceptable):

Summary	Definition
Access only when logged in	A user shall not have access to non-public functions or information if they have not logged in or have logged out.
Limited casual visitor access	A casual visitor to the Web site (who has not been authenticated as a customer) shall have only limited access. They shall not be able to even initiate any function that involves money (such as placing an order).
Employee maintenance access	The ability to maintain information about an employee (add, change, and remove) shall be limited to members of the human resources department.
Employee access to customer inquiries	A customer service operator shall be able to access all inquiries accessible by customers, to view exactly what any selected customer would. The operator shall be able to do this only for customers of the company for whom the operator works. It is not acceptable to achieve this by having an operator log in as the customer. The purpose of this is to allow an operator to explain to the customer any information they have difficulty understanding, and to see exactly what the customer sees: identical data in an identical format.

Category 3, actions within functions:

Summary	Definition
Customer maintenance	Each type of action that can be performed within the customer maintenance function shall be subject to separate access privileges. "Type of action" shall include changes to address and credit limit.

Category 4, data:

Summary	Definition
Service bureau has access to client companies	Employees of the service bureau shall have the same access to each of its client companies as the employees of those companies themselves have.

Summary	Definition
Company financial information access	Company financial information shall be accessible only by members of the finance department and senior managers. For the purpose of this requirement, "company financial information" means figures pertaining to the overall performance of the company, and other information of an accounting nature; customer order and payment information are not classed as company financial information.
Inquiries and reports not to show inaccessible data	No inquiry or report shall show data to which the current user does not have access. Where appropriate, inaccessible data shall be "filtered out." This shall be done in such a manner that summary information (totals, averages, and so on) are consistent with the data that is shown.

Categories 2 and 4 combined, both functions and data:

Summary	Definition
Configuration maintenance access	Only nominated employees shall be allowed to modify configuration parameters, and then only in areas expressly designated. For example, a finance manager might be allowed to modify only finance-related parameters.
Company can run agents' reports	The company shall be able to run for its own use all the reports available to its sales agents, to show information for any selected agent.

Category 5, limits:

Summary	Definition
Customer refund limit	An employee shall be able to approve a refund to a customer up to (and including) the refund limit set for them.

Category 6, time:

Summary	Definition
Initiate transactions only during nominated hours	An employee shall be able to initiate transactions during nominated hours of the day. It shall be possible to specify these hours of the day for each employee, but if they have not been set for an employee, a configurable default range of hours shall be used.

Category 7, environment:

Summary	Definition
Employee remote access	An employee shall be able to access the system from outside company premises if authorized to do so (and not otherwise).

Category 8, strength of authentication:

Summary	Definition
Reduced access by employee without smartcard	An employee who has been issued a smartcard but who logs in without it shall not during that session be able to initiate or approve financial transactions.

Category 9, transference:

Summary	Definition
Log data access as per original data	Access to data stored in a log shall be restricted to at least the same degree as access to the original data itself. For example, if a user is allowed to see customer details only for one company, they shall not be able to view details of a log entry about a customer associated with another company.
Query information access	Information that satisfies a query by a user shall be filtered to exclude anything the user does not have permission to view.
Document comments access	Access to comments made on any document shall be subject to the same controls as the document itself.

Category 10, operational rules (of which there is a further example in the "Cannot Approve Own Action" section in the approval requirement pattern):

Summary	Definition
Cannot extend own authority	No user shall be able to modify their own access privileges. In particular, no user shall be able to extend their own privileges.
View own orders only	The system shall permit a customer to view only orders that they placed, not orders placed by other customers.

Category 11, blanket bans:

Summary	Definition
No unauthenticated, uncontrolled, or blanket access	Every activity required for the normal commercial operation of the system shall be subject to all the access control requirements in this document. This requirement demands in particular: 1. Low-level access to any database shall not be required. This includes SQL queries and any application that permits access equivalent to SQL queries. 2. Command line access shall be restricted as per the next requirement. If or when this requirement is satisfied by all systems running on a particular server machine, then any of the above types of access described as "shall not be needed" can and should be toughened to "shall not be permitted." (Even if, for the operation of other systems, users must be granted uncontrolled access of the sorts this requirement is intended to prevent, policies should insist that those mechanisms not be used in relation to *this* system.) This requirement does not apply to steps necessary to rectify serious system problems, reconfiguration, or installation. Nevertheless, in those cases, this requirement should be relaxed only as far as is necessary to get those jobs done. (To compensate for the reduction in security in such situations, it is recommended that additional manual controls be applied—such as closely supervising staff when they undertake these tasks.)

Summary	Definition
No command line access	No command line access to the operating system of any server on which the system runs shall be granted to anyone involved in the day-to-day operation of the system. This includes anything equivalent to command line access, such as uncontrolled applications and the copying in of arbitrary programs, scripts, or other files. The motivation for this requirement is to force every action needed to operate the system to be available via a function whose use can be controlled and recorded, and to minimize the number of people who need command line access. This requirement does not apply to steps necessary to rectify serious system problems, or to install, upgrade, or reconfigure the system. (Nevertheless, in those cases, this requirement should be relaxed only as far as is necessary to get those jobs done).

Extra Requirements

None, beyond those common to both types of user authorization requirement.

Considerations for Development

Specific authorization requirements can be varied, and the ways to implement them can be equally varied. Deal with each one on its merits. Don't seek a grand mechanism that can solve them all: they might be too diverse.

The biggest decision is whether configurable access control is worth implementing, even if there's no explicit requirement for it. You can't expect a requirements specification to specify every little detail of who's allowed to access what. Judge for yourself how big the gaps are. Maybe there's an unspoken expectation of configurable access control; if so, bring it out into the open and resolve it quickly.

Considerations for Testing

Testing a specific authorization requirement is reasonably straightforward: can that kind of user access whatever it refers to? Are all other kinds of users prevented from accessing it? If a user is able to delegate their authority to another user, test that a delegatee is authorized whenever the delegator is. Change the kind of a user (if that's possible, for example by assigning them a different role), and test that their authorizations change accordingly.

An invaluable tool when testing authorizations is for all rejected authorization requests to be chronicled (logged). When reviewing requirements, check that this feature is included. If not, insist on it.

11.5 Configurable Authorization Requirement Pattern

Basic Details

Related patterns:	Extends user authorization; refers to authorization access, chronicle
Anticipated frequency:	Up to two requirements
Pattern classifications:	Affects database: Yes