Towards a Framework for Security in eScience

Andrew Martin, Jim Davies, and Steve Harris
Oxford University Computing Laboratory
Wolfson Building, Parks Road,
Oxford OX1 3QD, UK.
Email: firstname.lastname@comlab.ox.ac.uk

Abstract—This paper describes an approach to the formulation and classification of security requirements in eScience. It explains why it is untenable to suggest that ‘one size fits all’, and that what is an appropriate security solution in one context may not be at all appropriate in another. It proposes a framework for the description of eScience security in a number of different dimensions, in terms of measures taken and controls achieved. A distinctive feature of the framework is that these descriptions are organised into a set of discrete criteria, in most cases presented as levels of increasing assurance. The intended framework should serve as a basis for the systematic analysis of security solutions, facilitating the processes of design and approval, as well as for the identification of expectations and best practice in particular domains. The possible usage of the framework, and the value of the approach, is demonstrated in the paper through application to the design of a national data sharing service.

I. INTRODUCTION

eScience projects necessarily work at the extreme of what is feasible with current technology, and issues of security are not always uppermost in the scientists’ minds. However, these projects often involve sensitive and valuable data, and scientists or their employing organisations may be carrying significant liability.

Projects have been enjoined to take a risk-based approach to security [1], and this appears to be happening [2]. However, the risk assessments are relatively ad hoc, and may produce quite different results for two projects that are the same in every essential respect.

Project architects are often not security specialists, but are expected to choose from a plethora of overlapping, often experimental solutions [3]. These solutions are themselves under development, and the assurance delivered in context can be difficult to determine. Projects subject to regulation—for example, in eHealth—are required to present risk assessments to oversight bodies who may not be in a position to judge the appropriateness of the technical controls proposed [2].

Technologies and processes for security continue to develop and mature. Some ideas—such as trusted computing—hold great promise for improved integrity and confidentiality controls in eScience applications. To determine application areas which might usefully benefit from such technologies, it is useful to define the space in which they operate, and the categories of threat they serve to mitigate.

Our purpose in this paper is to propose a number of aspects or areas of security relevant to eScience projects, and to describe for each a number of discrete bands or levels of achievement. These are generally ordered from low (least resistant to attack) to high (most resistant), but this is not a linear order in every case. We do not suggest that every project should aim for the highest level in each category—in some contexts this will be infeasible, or at least inappropriate.

Instead, we suggest an extensible framework of aspects and expectations, a collection of considerations with a list of measures and corresponding achievements, organised where possible in terms of existing standards. In some cases, the discrete achievements are not hierarchical, but instead bilateral alternatives which are not directly comparable.

The framework can be used in the design of a project architecture, and in the selection of specific components and solutions. The intention is that solutions could be rated, or understood, against the criteria specified here. Our aim is not to be prescriptive, but to provide a basis for the explicit consideration of a range of important issues—within a project, and in conversations with third parties. We aim also to produce a useful contribution in terms of ‘best practice’ guidance.

A. Related work

A key inspiration for this approach is the NIST Electronic Authentication Guideline SP800-63 [4]. This defines four levels of identity assurance, the technical and procedural measures needed to achieve each one, and the forms of threat which are mitigated by each level of achievement. A project or organisation may define requirements, and conduct risk assessment, with reference to these levels of assurance. A service provider may characterise the level of assurance offered in the same terms, and present evidence that this has been achieved.

Example 1: The UK e-Science Certificate Authority, and its peers in other countries, is designed to comply with level three in the NIST authentication framework. This means that: government-issued photo ID must be produced in order to enrol, and secret keys associated with certificates issued by the CA should be protected by passwords, and used only in suitable contexts. By contrast, the adoption of Shibboleth by UK universities is approximately at level two in the NIST authentication framework. Enrollment is typically bound to University identity regimes, and authentication is often by means of a password-based protocol. Relying parties, such as the participants in the UK National Grid service, must decide whether level two authentication is sufficient for access to NGS resources, or whether stronger authentication is needed.
The ES-LoA project at the University of Manchester [5] has reviewed means by which similar levels of assurance can be defined and implemented for the e-Infrastructure community. Its report reveals a high degree of aspiration (towards well-defined levels concomitant with risk, and towards the implementation of systems capable of performing at level 4), coupled with a startlingly low degree of current compliance (no identity providers surveyed met even the detailed requirements of level 1 on password selection and other criteria).

The closest general purpose regime we are aware of is the Common Criteria [6] for security evaluation. This is focussed upon high assurance systems, often deployed in the protection of national security. The approach taken is to define a product (‘target of evaluation’) and a protection profile, and to submit the former for evaluation against the latter. Certification is possible at a number of different levels, depending upon the extent of the demonstration of compliance. The Common Criteria have been criticised as lacking in real-world considerations, and as an exercise in bureaucracy and blame-shifting, rather than genuine security enhancement [7]. Our present approach does not imply any third-party evaluation of compliance, though this is perhaps a possible future development.

Another perspective comes in the security guidance of ISO/IEC 27002 [8]. This standard defines—not prescriptively, though in a manner capable of being audited—security controls which may be relevant to a wide range of projects. Its approach may be regarded as a “check-list” of items to consider; the final selection of controls resting with local project experts. Our framework can be seen as complementary to the 27002 guidance. Similarly, a NIST standard SP800-53 [9] details a wide range of security controls from which product and system designers can select for achieving a particular outcome. Our purpose is more specific, but not at odds with this standard.

B. Objective and Outline of paper

Good security involves process as much as product [10], and we have tried to avoid drawing our headings too narrowly. In Section II we describe the aspects under consideration, and the levels of assurance associated with them. The aspects are: identity assurance; attributes, authorization, and accountability; crypto selection and management; processing of personal data; anonymization and pseudonymization; isolation; user training/sophistication; code quality/process maturity; rights management; provenance and results assurance.

Where possible, suitable technical measures are defined. It could be argued that the resulting framework mixes requirements with designs and solutions: this would be the case for a ‘blank sheet’ development, but almost every eScience project will be constrained by existing and related architecture and components. Further, we identify a sample of threats that would be defeated by a sound implementation, although space prevents any more comprehensive characterisation.

The contribution of this paper is to suggest a separation of concerns into orthogonal or at least clearly distinct issues, and to provide a set of categories or levels of assurance against each. By delineating discrete categories, we hope to advance the state of the art, drawing attention to technologies and approaches that are inherently stronger, as well as to the trade-offs, risks, and costs involved. We hope to provide a basis for the evaluation of combinations of solutions, working together to mitigate particular threats; we hope also to assist designers in adopting and justifying architectures and approaches that are consistent and economical in their treatment of specific threats. The framework is illustrated though application to the design of an existing project, a national service that will facilitate data sharing between research programmes.

II. AREAS FOR SECURITY

A. Identity Assurance

1) Problem statement: The NIST Electronic Authentication Guideline SP800-63 is our inspiration and addresses identity assurance in more detail than we could hope to present here; any reader whose interest is engaged by our brief summary of the different levels may like to explore the full document [4].

2) Levels of assurance:
   - L1 self-asserted identity; may be pseudonymous; authentication using password or PIN based crypto protocols
   - L2 photo ID or local organisation (University, etc.) enrollment; stronger protocols for better attack prevention
   - L3 photo ID and better checking; authentication using cryptographic keys and password protection
   - L4 multiple pieces of ID, cross-checked; hardware-backed storage for cryptographic keys

3) Protections Achieved: Adapted from [4, Table 3]:

<table>
<thead>
<tr>
<th>Protect against</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-line guessing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Replay</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Eavesdropper</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Verifier impersonation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Man-in-the-middle</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Session hijacking</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

4) Discussion: The reliability of these levels of assurance depends upon many factors, including the quality of software design and implementation—considered as a separate aspect below. It depends also upon the provision of audit procedures, to ensure that stated policies are indeed being followed. An independent observer can make some evaluation of the formal protocols in use, but the evaluation of the rigour of the associated procedures requires privileged audit access. The associated cryptography must of course be of a suitable standard, and technical concerns surrounding the appropriate storage of keys must be addressed. Moreover, the usability of the identity verification systems—in terms of their comprehensibility, robustness, efficiency, and graceful failure—is of critical importance.

B. Attributes, Authorization, and Accountability

1) Problem Statement: Policy-based decisions, typically relating to authorization, are made on the basis of attributes,
such as employer and employment status, membership of project teams, roles, external subscriptions, security clearances, and so on. But are those attributes appropriate? Are their values correct, or current? Who says so? Is the attribute authority actually authoritative? Is the way in which these attributes are being used consistent with their intended meaning, from the perspective of the authority?

2) Levels of Assurance: For each individual attribute, there are a number of levels:

L1 Authority uses ‘best efforts’ to present accurate attributes; there is no auditable process. Procedures may involve self-certification by users, manual collation of lists from other sources, etc.

L2 Authority uses a transparent, auditable process to collate attributes from authoritative sources. Publishes procedure, including length of time between updates (lifetime of attributes) . . .

L3 Authority is the authoritative source; assertions are necessarily accurate (by definition) at the timestamp when the assertion is made. Authority may issue guidance to relying parties about assertion lifetimes.

3) Discussion: The management of access control policies is also of interest: even when these are well-established, the attribute data on which they depend may be of variable quality, as our list of levels suggests. Some discussion has arisen on the subject of who controls what: for example, whether the resource owner, the project PI/manager, or some other party, has to participate in the setting and inspection of quotas.

C. Crypto Selection and Management

1) Problem Statement: Almost every project will have some need of cryptographic technology for ensuring the confidentiality or integrity of data. A great majority, however, can rely upon the normal, default implementations in commodity software distributions—although they may need to ensure that these are properly configured. Projects with data of a sensitive nature, or where the confidentiality requirement is particularly long-lived, may need to select cryptographic tools more carefully.

Example 2: The secrecy of some patient data is sensitive throughout the patient’s life.\(^1\) Good quality advice is that many of the cryptographic algorithms in use today should be considered strong enough to protect secrets for around 30 years. A risk assessment might therefore conclude that such data should not be sent across the Internet using a commodity cryptographic suite, unless other mitigations are in place.

A similar problem arises with the integrity of archival records: a digital signature made today may be forgeable in 30 years’ time. Some archives must put in place a suitable mix of protections (holding records ‘in public’, using unalterable media, ‘rolling forward’ signatures by re-signing within the lifetime of the previous signature).

Moreover, any regime which requires long-term data storage must define appropriate key management and access control procedures: for most purposes these must be role-based and should assume that the technical means of authentication and authorization will change over time; issues of key ownership and escrow are important also.

Long-term crypto should be defined with reference to best practice. Detailed guidance is to be found in the NIST Recommendation for Key Management [11], although 2030 is the furthest it is prepared to go in any of its recommendations; the highest level defined below might thus be judged fanciful, but it does represent a legitimate aspiration.

Cryptographic controls may apply to the transmission of data (over networks, on memory sticks and discs), or to data at rest (in filing systems, databases, backup tapes). For the latter, to say that ‘it is encrypted’ may be insufficient. We should distinguish between data that is: user-managed (locally, or on server, or in cloud); curated; encrypted (by service; keys for access negotiated with user); or encrypted by end user (and so uninterpreted by service).

2) Levels of Assurance: Here, our concern is with the protection of data, rather than authentication and authorization.

L0 No cryptography required; there is no plausible threat to integrity, and data secrecy is not at stake.

L1 Standard cryptographic functions and libraries are used, with commonplace key sizes, software keystores, and software random number generation (hardware devices for efficiency, e.g. encrypting network cards, but not necessarily for increased security); encryption for secrecy; timestamped signatures for authenticity.

L2 As for level 1, but with hardware keystores and hardware key/random number generation; procedures for key management, token-based key storage.

L3 Specialist cipher suites, for very long term (exceeding 30 years) protection and/or very high assurance; associated procedures for curation of data and keys.

A separate issue for each of these (apart from Level 0) is whether the cryptography is (a) undertaken by the service provider — the long-term data store, for example, or (b) undertaken by the data owner (locally to the data collection point, or within computational jobs distributed elsewhere)

3) Discussion: The levels protect against increasingly well resourced attackers and/or increasing sophistication of attack. Token-based authentication may be adopted without token-based key storage, but the converse is not generally sensible. Anecdotally, it seems that an alarming proportion of eScience projects are presently at Level 0, and that the sole use of hardware keystores is in CA operations and other high-value assets such as proxy key stores and credential translators.

D. Processing of Personal Data

1) Problem Statement: The definition of, and required protections applied to, personal data vary according to different jurisdictions and contexts. We assume a stratification with well-defined concepts of (and handling requirements upon) personal data and sensitive personal data: with more or less...
stratification, the number of relevant levels might grow or shrink.

2) Levels of Protection:

L0 No personal data is processed.
L1 Simple kinds of personal data are held and used (probably subject to regulation by government authorities).
L2 Sensitive personal data are held and used, subject to regulation and also requiring explicit consent from the data subject.
L3 Sensitive data is processed as at L2, and this is additionally subject to detailed regulation by an ethics committee or similar body.

3) Discussion: These levels are different in character from those in the preceding sections: they characterize the project or system, rather than the controls which are to be imposed. They may themselves give rise to requirements in other aspects of this framework (though due to differences in interpretation and context, this may not be in a particularly systematic way), as well as requirements upon data retention and assured data destruction.

E. Anonymization and Pseudonymization

1) Problem Statement: Some projects work with pseudonymized data, or may even see a requirement for completely anonymous data. In practice, different degrees of pseudonymization may apply, together with a range of mechanisms for re-association. At the extremes we have ‘true anonymity’ and ‘accurate association’ (the research literature suggests that the latter may be infeasible or even not meaningful), and in the middle varying degrees of likelihood or membership of varying set sizes. We may also consider the degree of effort involved in reassociation: trivial processing, brute force, or statistical methods upon the data. This suggests at least a two-dimensional problem space, with continuous variables on both axes.

2) Levels of Assurance:

L1 obfuscation of identifying data (good for preventing accidental disclosure, where the data processor might inadvertently discover personal data relating to a colleague, friend, or family data)
L2 reversible mapping of identifying data (mapping should be keyed; the key tightly controlled: status of such data may be ambiguous with respect to legal requirements)
L3 the identifying data is replaced by genuinely random keys; the mapping from the identifying data to the new keys is held as a lookup table by a trusted authority (similar remarks to the above).
L4 genuinely irreversible mapping of identifying data.

3) Discussion: Anonymization must take account of data semantics (for example, location, date of birth) and concerns arise with longitudinal data release and cross correlation among data sets. The size of the dataset relative to the population is a key consideration. There is a considerable amount of research already in this area, and it may be incorporated in this framework. Finally, we should observe that, for some purposes, L4 will be inappropriate—if there are circumstances that might demand reassociation.

F. Isolation

1) Problem Statement: A significant amount of investment in security relates to technologies for isolating users, processes, software, or data from one another; this is of particular importance in grid and cloud settings, as well as in “high throughput” or “public resource” computing models. If enabling software (operating systems, grid middleware, network firewalls) could be seen as perfect, this would be less of a challenge, since one of the principal tasks of such software is to provide a measure of isolation. Even then, however, the design of policies to be enforced by those systems must take account of the possibility of attack via any of the components. Since many systems are intended to allow remote users access to computational resources—for arbitrary jobs or services—a suitable ‘least privilege’ can be hard to enforce.

In this section, we use the term ‘job’ to describe a generic task undertaken by a system, even though grid and/or cloud systems are seen increasingly in terms of services, often assembled in workflows. Services raise separate, different security concerns, but for the purposes of classifying isolation they may regarded as long-running batch jobs.

Example 3 (Data Enclave): Applications with large collections of highly sensitive data frequently implement data enclaves. The database is loaded into an isolated system, contained within an access-controlled room. Researchers who have entered into appropriate agreements travel to the location and are given access to the room, with the caveat that any data to be removed, such as aggregated results or query outputs, must be viewed and approved by a person or committee acting as a data controller. Virtual implementations—necessarily with less assurance because the user is not physically supervised—involve remote desktop services, but with export capabilities (cut and paste, email, file transfer) disabled; the intention is that the only export channel should be via a data controller.

Any non-trivial software suffers from vulnerabilities, and jobs may exploit these for a variety of reasons. Isolation of data at different levels of classification has long been a goal of government and military systems, involving the implementation of models such as Bell–La Padula. The assurance achieved can be less than might be expected, given the size of the investments made, partly because designs must take account of esoteric features such as covert channels.

Example 4: When using Condor [12] there is a mutual isolation challenge: the job executing on the desktop Condor client must be prevented from interfering with the interactive user; the interactive user must be prevented from stealing the input (data, executable, and credentials) or output of the Condor task, and from tampering with the output.

Some protections are easier to subvert than others; some are more expensive to implement; some are more intrusive. Some are inherently more sound, some are better implemented,
and some are observably the subject of regular vulnerability reports. It is important that data owners, project architects, and resource providers are able to make an informed selection.

2) Levels of Isolation:

L1 Process-level isolation only. Each job runs as an executable within a general-purpose account to which other user(s) have access. It has access to the user’s capabilities and data.

L2 Isolation as afforded by operating system accounts: either transient accounts which are eventually recycled (with a risk of temporal data compromise), permanent accounts, or a one-account-per-service model.

L3 Sandboxed execution. This offers protection against rogue processes that could otherwise interfere with user processes and data.

L4 Whole system virtual machines:
   a) “Type II” hosted VM: offers substantial separation, but ultimately weak against an attack upon the VM from the host.
   b) “Type I” potentially very strong isolation.
   c) Trusted virtualization. Use trusted computing technologies [13], [14] to provide a strong guarantee of which VMM is in use, and of the signature of each job/service.

L5 strongly (physically) isolated platform
   a) using a firewall or equivalent mechanism to manage incoming requests
   b) using complete physical isolation (‘air gap’): stand-alone (non-networked) system or entirely isolated network.

3) Controls Achieved: The protections defined above cannot truly be said to ‘prevent’ particular attacks, but instead make them less feasible, or less likely to arise in practice. A more comprehensive account of the framework would include a breakdown of threats and mitigations.

4) Discussion: It is difficult to achieve a precise matching of risks to protections for this aspect of security: most of the protections are intended to address the possibility of unknown code exploiting unknown vulnerabilities.

Protections suitable for a data centre are not necessarily suitable for desktop grids, and vice versa. Physical access to devices tends to imply the ability to subvert them, but doing so can—in theory—be rendered difficult, expensive, or easily detectable.

Isolated networks at L5 may provide an illusory level of protection: as the network becomes larger, it takes on characteristics of the internet itself—in terms of the number of people who have access, their motivations, their level of technical expertise, the scope for the introduction of rogue software, and the difficulty of auditing all access. Firewall configuration is seldom perfect. Truly isolated systems must be subject to strong procedural controls, lest they give way to ‘sneaker net’ technologies. Trusted virtualization is decidedly only at the prototype stage at present.

G. User training/sophistication

1) Problem Statement: The socio-technical aspects of security are too often overlooked. The environment in which a system operates has a clear bearing upon its design, upon the assumptions made, and upon the degree of security achievable. The level of training, IT sophistication, and security awareness of the user community impacts: the design of interfaces; the extent and complexity of controls—for example, in terms of the acceptability of token-based authentication; the likelihood that some person with legitimate access will attempt to abuse that access—for nefarious purposes, or simply to do their job; and the perception of the likelihood of being caught and/or punished.

2) Levels/Kinds of User:

L1 untutored general public
L2 low-stakeholder academic (student, etc.)
L3 higher-stakeholder academic: researcher who has intellectual property, professional or personal investment in the item under consideration
   a) with technical sophistication (“hard” sciences, typically)
   b) with less technical sophistication
L4 corporate user with high-grade intellectual property protection concerns, with pre-existing training to match.

3) Discussion: A user’s motivation to complete a particular task at a particular time is also a key consideration: a high-stakeholder might decline to break procedures, but might either (a) call upon support staff to solve a problem or (b) tacitly allow subordinates to do the breaking. Those with a lesser stake but a short deadline may well bend the rules—often via identity-sharing rather than more sophisticated methods. Many systems will have a mixture of users, and we should consider those with the least level of training/sophistication.

H. Code quality/Process Maturity

1) Problem Statement: It is well-known that even the best designed systems can have substantial vulnerabilities. No significant pieces of software exist which have no vulnerabilities whatsoever; some limited-scope and/or high assurance systems may claim complete correctness, but many of those claims are difficult to sustain. A multi-tasking operating system with extensive middleware certainly could not reasonably claim to be flawless.

Good engineering practice tends to minimise vulnerabilities, but is often lacking in the development of research-level code: because the skills required to build robust code are inaccessible in context; because the development approaches adopted are inappropriate [15]; or because there is no budget or reward for bringing the code to production quality.

Further, it is widely accepted that in order to be trustworthy, software may need to be frequently updated, in the light of vulnerability reports. This requires a level of ongoing support very often missing in the scientific community. Rather than make a fresh definition here, we incorporate the CMMI level descriptions for process maturity [16].
2) Discussion: Although higher levels of process maturity are no guarantee of security, lower levels make it more likely that code quality is low, and that vulnerabilities are present.

I. Rights management

1) Problem Statement: Rights management relates to the attachment of policies to mobile data, so that access controls can be advised/enforced no matter where that data is processed. Most would see this as a development of mandatory access control; it can also be viewed as a distributed implementation of policy decision and enforcement points.

The problem has several dimensions:

- the expression of such access control policies: this is a natural extension of other systems of access control;
- the calculation (or other derivation) of policies to attach to the output of processes/jobs, based on the policies applying to the input, and the semantics of the task; and
- the strong enforcement of policy decisions.

This is a contentious aspect of security, not least because media companies have already used rights management technologies to construct unpopular business models.

2) Levels of Rights Management:

L0 No rights management enforcement. Parties with access to data are expected to process it in line with agreements, and not to retain it too long. This may be audited manually, but no technical measures are in place to prevent departure from the policy.

L1 Advisory/weakly-enforced sticky policies are used, and data comes with attached privacy/confidentiality policies. Data users are assumed to be intending to comply with the policies, and software libraries are supplied which help them to do this; these could be circumvented by a determined insider.

L2 Strongly enforced sticky policies/DRM policies. Software is designed to permit distribution of data only to hosts which will enforce the policies, and executes on trusted platforms.

L3 as level 2, but with appropriate backward/forward secrecy for virtual organisations.

3) Discussion: Level 2 may be suitable for the distribution of licenced content (such as licences for Ordnance Survey map data); Level 3 might cover the restriction of a broader class of data to the members of a particular virtual organization; Level 1 works well in regulated contexts where most actors are assumed to be intending to comply with the policy.

Some will argue that level 2 will always be unattainable: that any enforcement regime will be defeated by determined attackers—many precedents suggest that this is the case. On the other hand some very widely-deployed media protection systems—for example, BSkyB’s encryption system—have no known class attacks, suggesting that with appropriate design and control of the system, some strong protections are feasible.

Others will argue that level 1 is not security per se, since it does not deal with the defeat of any attacks.

J. Provenance and Results Assurance

1) Problem Statement: Accurate provenance information for results is of increasing importance in sciences where results are based upon computational analyses. Its generation and preservation is in some degree a dual problem from that of rights management — the one deals with the confidentiality of data wherever it may go; the other deals with the integrity of data wherever it may have come from. Although many projects survive with no provenance data, and no guarantees, regulatory contexts increasingly require provenance data, and it will eventually be necessary to demonstrate data accuracy, integrity, and derivation.

2) Levels of Assurance:

L0 no attempt made to verify computational results; no attempt made to record/trace which computations took place;

L1 provenance is addressed at the level of the overall process (and might be recorded at the level of the data set); guarantees of accurate processing come from:

a) processing in an assumed trustworthy context (e.g. a particular grid or cloud service);

b) duplication of computations, perhaps coupled with statistical analysis and/or majority voting [17];

L2 provenance data attaches to an individual datum;

L3 as with level 2, but strong assurances of accurate processing come from digitally signed processing in a demonstrably trustworthy context (see Isolation (Section II-F2.L 4c above).

3) Discussion: For Level 1 when processing is outsourced—for example, to a pool of public participation hosts—the integrity of results depends upon whether an attacker has motivation to subvert them, and the capability to produce plausible (but erroneous) results. The means for verifying the accuracy of results from such processes may vary from statistical methods to full duplication of tasks [17].

III. Case Study

As an example of how the framework may be applied, we will consider the design of a national data sharing service: the UK Medical Research Council (MRC) Data Support Service (DSS). The MRC supports a number of population studies: long-running, longitudinal cohort studies looking at, for example, child and maternal health, genetic markers for disease, and health and social factors in aging.

Each study collects detailed information, much of it sensitive in nature, on large numbers of people (a typical study will have several thousand participants); a key aim of the DSS is to facilitate appropriate, uniform access to data, allowing researchers to explore associations between the data collected in different studies.

A. Identity Assurance: The first aspect considered, that of identity verification and authentication mechanisms, reveals an essential characteristic of the service. The study units take great care in verifying the identity of those who will be involved in data management (L3), and will authenticate to
either L2 or L3. It is impractical, however, for the service to verify the identity of researchers with general access to the system to more than L1, or to employ more than L1 authentication. This can serve as valuable input into discussions of trust between different stakeholder groups.

However, the table of protections achieved takes matters further, and makes the point that although L2 may be acceptable for data managers within their units, it is certainly not acceptable if these data managers are to make use of the external resources of the DSS in handling sensitive data. If this is an aspiration for the service, then appropriate tools and authentication services must be offered; furthermore, it may be that some services, such as annotation services, cannot be provided centrally, and that a federated approach must be adopted instead. In this respect, applying the framework may raise fundamental questions about the scope and nature of the development.

B. Attributes, Authorization, and Accountability: This aspect of security is easily confused with identity management, and to have it as a separate concern is particularly useful. In considering whether user A should have access to the privileged, detailed, or sensitive data of study B, we can now safely ignore the related question of user A’s true identity. In the DSS, any access to privileged data is granted by the data manager associated with the study in question. The service will reach level L2 of the classification, but not L3.

This should not be controversial, as previous data sharing procedures can be seen to be also at L2: units cannot guarantee that the information they hold on someone they have shared data with is still current. In applying the framework, however, it was clear that further guidance would be useful—perhaps to the extent of an additional aspect or classification—with regard to auditing, revocation, and policy enforcement.

C. Crypto Selection and Management: At first sight, this aspect appears to be of limited interest: data management within the DSS will rely on standard packages, and the cryptography employed will be that which comes with the package selected. However, the DSS is the obvious candidate for taking on the role of data management when studies and units close, or when investigators and data managers retire. Many of the studies involved have data that is more than 20 years old, and one of the studies started collection in 1947. The specified levels of assurance then give cause for thought: studies currently hold data at L1, with some practices characteristic of L2; the designers need to reflect upon whether they should aim for L3, and whether they should be assisting the study units in doing the same.

D. Processing of Personal Data: For the DSS, all data and metadata, apart from the core user data, has been considered by ethics committees, and all information is collected with explicit, informed consent. New standards and procedures are likely to emerge with regard to consent in long-running health and population studies, particularly those with associated tissue and blood samples, and the framework can make reference to these.

E. Anonymization and Pseudonymization: This is an area of obvious concern for population studies, and each study unit has existing procedures. For the most part, these procedures correspond to L2: the mapping between keys and identifying information is tightly controlled, but unlikely to be held by a separate authority. The criterion for L3 does suggest a plausible, future role for the DSS: not least because it is preferable to adopt and maintain a separate mapping for every data sharing instance. L4 may not be desirable in this case: there may be an ethical imperative to identify a participant in order to make them aware of a health condition; also, study units will wish to use data derived from shared subsets to enrich their original data set.

F. Isolation: The mappings between identifiers in the original data, and the anonymous identifiers used in shared data sets, are typically held on computers that are not connected to the network. This “air gap” approach to security has clear advantages, but relies on error-prone, manual processes that do not scale with increasing numbers of data sharing requests. So although current practice is at L5, a successful deployment of the DSS will require interoperation across firewalls, and is likely to push the level of isolation downwards, perhaps to the level (L2) employed for most activities within the units.

G. User training/sophistication: All of the study units have a strong culture of responsible data handling, and their processes are designed to be executed by staff with a high level of training (L3). This is an area in which the common infrastructure—the economy of scale—and the degree of automaton afforded by the DSS will improve or safeguard existing standards, in reducing the risk of inappropriate delegation to untrained staff (at L1 or L2).

H. Code quality/Process Maturity: The core information systems used within the units are developed to CMMI L2. There is no doubt that the same expectations will be applied to the implementation of the DSS. The DSS will hold schemas for each dataset, and will allow users to construct a “shopping cart” specification for a data sharing request; it is thus perfectly feasible for the service to generate query implementations and workflows to match each request, reducing the need for bespoke code development and increasing overall quality.

I. Rights management: In the context of sharing sensitive data, rights management is an issue of particular interest. At present, when a subset of data has been released, the unit no longer has any automatic control over its usage. However, the weight of existing technologies and practices make it unlikely that much progress will be made in this area in the near future, and the DSS can thus provide only L0 assurance.

J. Provenance and Results Assurance: Every item of data available via the DSS will be associated with detailed, semantic metadata, describing not only the context of its collection but also its relationship to other data items. Models of studies, datasets, and sampling events (questionnaires, clinical surveys) will be complemented with references to a document repository holding, for example, study protocols and operating procedures.
The intention is that a metadata registry and associated services will support the annotation not only of data, but also of the software involved in the acquisition, processing, and storage of data. The goal is to ensure that the derivation of any data produced by the DSS is immediately accessible; that this goal is achievable demonstrates the exemplary measures taken by the units involved to document their data in a logical, computable fashion. This corresponds to L1, with elements of L2.

IV. CONCLUSION AND FUTURE WORK

This paper attempts to cover the major security issues raised in a series of eScience workshops. There will, of course, be other aspects to consider, as technology evolves and new application areas are explored. The partitioning of issues into separate aspects is based upon discussions in those workshops, together with direct experience of security issues in several large eScience projects. The extent to which it seems appropriate, or natural, may depend upon the application area involved. The identification of discrete levels is absolutely essential, to avoid ambiguity about what is being claimed and achieved. The extent to which these are security issues, as opposed to broader issues of system functionality, is open to further interpretation.

The framework presented is clearly in need of further elaboration, something that is best accomplished through application to a wide range of projects and systems. A more complete description of the framework will fully describe the protections achieved and threats mitigated by each level of achievement. This could help to match solutions to projects, and to identify ‘clusters’ of application areas with similar security concerns. However, our initial observation is that there is no simple progression from ‘low assurance’ projects to ‘high assurance’ ones.

For some of the aspects considered here, higher levels of assurance are associated with recognised, open research challenges. We hope that by defining interesting areas we may stimulate discussion on how to achieve such heights. Furthermore, it is one thing to envisage a higher degree of security, another to produce a proof-of-concept or prototype, and quite another still to produce a robust, working solution capable of wide distribution; further, applied research and new methodology may be required.

There is a particular characteristic of eScience systems that makes the promotion of security measures and concerns more difficult. In the early stages of projects, all that is required is that the system under development is ‘capable of being made secure’. At this point, efforts are focussed on the development of functionality: a greater concern is that the system is ‘capable of creating valuable data’. It is not until this data exists that the actual implementation of security measures becomes a pressing concern—when design decisions have already been taken, and the attention of the project architects may have moved on. It is our hope that a suitable framework of aspects, levels of assurance, and solutions will assist project teams in addressing security issues in greater detail, earlier in the lifetime of the project.

Despite this, we have deliberately avoided being prescriptive in our presentation: such a framework needs to be applied within the wider context of a risk-led process. An important factor in successful development is to pay just enough attention to each aspect of security: too little, and disaster may ensue; too much, and the project may be subject to unnecessary delays. Our purpose is not to reduce security design to an unthinking box-ticking exercise, but instead to facilitate constructive engagement with security issues, and thus to promote the cost-effective development of new, suitably secure eScience applications and systems.

ACKNOWLEDGMENT

The main part of this work was undertaken as part of the Theme on Trust and Security in Virtual Communities at the e-Science Institute in Edinburgh University. We thank the Institute for its hosting of the theme, and participants in the theme workshops for insights leading to the framework reported here.

REFERENCES