A Methodology for Integrated Networks and Applications

Joe Pruitt, 2004-22-09

Introduction

A fundamental problem in IT today is the lack of process and coordination between network and application teams in designing and deploying applications or services. This lack of coordination results in missed opportunities; better more flexible application scaling, more robust application high availability and disaster recovery, and stronger security.

In addition, if problems should arise with an application post deployment, lack of coordination can result in finger pointing during trouble-shooting and diagnosis. These types of events serve to further distance network and application teams.

With proper planning and design, utilizing both teams' unique capabilities, organizations can be better assured to lower application implementation risks while maximizing their infrastructure and their applications' high availability, scale, performance, and security.

The Role of Technology

Technology is an enabler and can even act as a facilitator of this type of integrated methodology but that's as far as it goes, the hazard is over relying on technology to solve all problems. A successful project is ultimately up to the people and processes surrounding the technology. It's often referred to as the three legged stool. Without one leg, the project can't stand. And, of course, technology decisions will need to be made based on the processes and people involved.

In selecting the underlying network technology and the design of the applications or services, some clear objectives need to be defined by the integrated team. Objectives are frequently based upon past lessons learned. For example, in the past we may have had problems with application performance where the application or service was connection limited or bound. This problem resulted in a lot of end user complaints regarding performance and even led to outages. So the first objective may be for the network to help provide flexible scale and better performance. This would be accomplished through virtualization of the applications which can help overcome the connection limit problem. This objective would then, in turn, drive other objectives. For example, for virtualization to work effectively, the network and application will need to have some type of communication indicating the health of the application. You don't want the network sending a user to an unhealthy application, node, or service. So the second objective would be that bi-directional health communication must take place between the network and application.

As you can see, these objectives would clearly have an impact on technology selection, and how the network and applications are designed, configured, and deployed. As with any endeavor, proper planning and coordination before a project starts will provide considerable savings in the long-term.

Role of People

Nothing can sour a team effort faster than a member or members who carry past baggage because there previously was no coordination between network and application teams. The people of a project are clearly the most important part so selecting team members judiciously is essential to success. I've seen creative ways to ensure that team members work together. They can be as simple as a written code of conduct that all members sign and agree to follow, to something as significant as reorganizing and aligning team members around a project or even a new "IT architecture" department. The important element is that the team has one document to refer to or one person to report to in order to resolve conflict.

The potential for conflict and misunderstanding can be helped tremendously by a sound process.

Role of Process

Process is a large part of any planning and organization for a successful project. An integrated team would follow common methodology currently used today in application development. The only difference is that additional stakeholders, the networking and security professionals, are involved upfront in the initial requirements gathering as well as design phase. (It's important to add that users of the application or service are also involved, which is an assumption we're making for the purposes of this document).

The requirements gathering phase is best conducted by stating what the problems are that you're trying to solve. Again, we're assuming that the business problems have already been well defined and that you've moved into the design requirements gathering phase. The problems and methodology we're referring to in this paper are centered on the core IT objectives of high availability, disaster recovery, performance optimization, security, and operational efficiency.

For example, say you're designing a web service based on SOAP and XML that is going to interface and automatically collect inventory information from a supplier within your supply chain. Cost is an important factor so instead of a dedicated leased line between you and the supplier, you want to provision or use an Internet connection. Security is an issue because the connections will be traversing through your firewalls, to and from the Internet, over port 80 (HTTP) or 443 (HTTPS). SOAP and XML are ASCII text, or human readable and thus insecure. A simple sniffer could grab sensitive data. The transactions therefore must be encrypted. Firewalls won't be of value because these ports are open and the firewall can't see port 443 encrypted traffic. In addition, you want to take measures that ensure the transaction is coming from your supplier and only your supplier, so some type of credentials need to be exchanged for authentication and authorization purposes and then checked against an LDAP database.

At the design phase, the developers could choose to design and develop an entire security model directly into the application themselves. This would be a mistake. The reason; it doesn't scale personnel or resources well; the developers would have to do this for each service or application that is implemented. Because of a security model's complexity and the need for specialized knowledge, mistakes can easily be made. If done independently for each services or application, as more come online, the model will become unwieldy and difficult to manage, maintain, and audit.

A better way is for the developers to involve the networking and security professionals in the beginning to see what network and security services could be leveraged, thus offloading the development team's efforts to the network team and centralizing these security functions. The benefits are better scaling of personnel and a centralized security model that is much easier to manage, maintain, and audit to prevent mistakes.

In order to execute on this type of methodology and achieve these benefits, a solid collaborative process between application, network, and security professionals needs to take place.

Steps for an Integrated Methodology

The following are suggestions of what a methodology, design and support process would look like for an integrated network and application team. These steps are merely intended as guidelines. As with any process or methodology, there can be several iterative steps in between.

1. Application and Network Design Requirements and Review:
An application and network design review would consist of reviewing application mock-ups or prototypes and uncovering the IT objectives and goals (high availability, security, performance optimization, operational efficiency) of the project. As part of discovery and requirements gathering, we've provided a sampling of questions that could be asked:

1.1. Will the application or service reside entirely behind the firewall or will it be distributed across multiple sites?
1.2. What are the application server and or middleware systems in use?
1.3. What database(s) and / or storage systems will be used?
1.4. Is disaster recovery necessary for this application or service?
1.5. What are the backup procedures and plans?
1.6. How many users are expected to access the application or service?
1.7. Will the users be inside the firewall, outside, or both?
1.8. Are the users company employees only or is it a broader base?
1.9. What types of access controls are required?
1.10. Is the data sensitive in nature, does it need to be encrypted all the time?
1.11. Is the application subject to problems if certain network conditions exist like jitter, latency, sizeable hop count, or packet loss?

2. URI / URL Hierarchy Design and Review:
If the application is a web-based application it's necessary to understand the URI / URL hierarchy. This will influence such things as the design and configuration of the network and servers. By conducting this type of review, scaling and virtualization determinations can be made. The following are some discovery questions that can be used.

2.1. Is there static content that would benefit from dedicated nodes?
2.2. Are there elements in the URI or URL that will need to be switched or persisted against?
2.3. Will URI or URL rewriting be needed under certain events?

3. Design for Session Portability and Scale:
Is the application being designed for session portability? This provides the greatest amount of flexibility for an application and the high availability network infrastructure supporting it but can be challenging unless properly planned. This is a critical element to determine because it will impact just how sessions will be directed by the network infrastructure, to what nodes, and what will be the optimal way to distribute those sessions among the nodes. This also has significant implications on back-end data and storage design. Some sample questions to be asked:

3.1. Are their unique variables that can be used in the header or payload of the data that can be identified by the underlying network in order to direct traffic to the appropriate pool of service or application nodes?
3.2. What are the key metrics that will influence the performance of the application or service? Can these be utilized to set the appropriate load-balancing algorithm?

4. Design for Stateful Applications and Scale:
For many applications stateless design or session portability is not possible or practical. If this is the case there are methods that can be used to optimize the application design to achieve the goals of high availability, scale, performance optimization and flexibility.

4.1. Can the client support such things as resets if a service or application is not available?
4.2. Is it possible for the client to store state information?
4.3. What are the unique parameters or variables that the underlying network can identify in order to persist a client to the appropriate node, application or service once a session or connection has been established; i.e. cookies, or unique service or session IDs?
4.4. What rules or policy language are needed in the network devices to support stateful applications?
4.5. What are the key metrics that will influence the performance of the application or service? Can these be utilized to set the appropriate load-balancing algorithm?

5. Design for Disaster Recovery / Business Continuity:
This analysis and planning would investigate what would be needed to support a geographically distributed architecture that could be utilized for disaster recovery, high availability, as well as performance optimization.

5.1. What are the necessary TTL (Time to Live) values to be set?
5.2. Once a request has been directed to a particular geographic location will the client session need to persist to the same location for the life of that session? This is to ensure application or service integrity.
5.3. Is geographic distribution going to be dependent on business criteria like business unit location?
5.4. Will remote users need to be identified and directed to specific locations where their application's or services reside? Will they need to be directed based on their proximity to a datacenter?
5.5. Are there network factors such as hop count, datacenter capacity, or packet loss that need to be factored into a global traffic direction decision?
5.6. Will you be using a 3rd party network for additional or occasional content or application delivery? If so, under what parameters should the 3rd party network be utilized; i.e. excessive datacenter capacity or location of client?

6. Application or User Directed QoS:
This step would plan for the design of how the application can be tagged to influence QoS based traffic direction for both inbound as well as outbound traffic.

6.1. Do you want to give different clients or users different priorities to applications or services?
6.2. Do you want to use the network to set QoS and ToS values in the packet?
6.3. Do you want the application to set values or QoS identifiers and have the network direct traffic based on such values?

7. Multi-Homing / Route Optimization Design:
In order to ensure that an application or service is always available and performing as best as possible, the connectivity of the application or service must be taken into account and planned. An analysis should be conducted on the amount of anticipated traffic and the best way to design traffic flow over multiple internal and/or external connectivity links.

7.1. Will the application or service need priority over other types of applications or services?
7.2. Does the application or service need its own dedicated set of links? Should these be shared under certain circumstances like unusually high load?
7.3. Is connectivity cost or performance a factor that should be considered in determining the appropriate link for the application or service?

8. Network Security Design Review and Plan:
Analyzing the needs of the application and the constituency which it serves will greatly influence network security implementations. For example, port 1434 may be necessary to open to allow remote SQL queries for clients. However, as evidenced by the recent SQL Slammer Worm, this can leave systems open to attack.

8.1. Do we need to open some ports for remote client access?
8.2. If so, are their unique variables in the session that can be identified and keyed off of to either allow or deny access?
8.3. Will IP filtering be needed?
8.4. Are we exposed to DoS attacks with this application or service?

9. Application Security Design Review and Plan:
Similar to network-level security this step seeks to support application-level security. This involves examining the use of application filtering, client certificates, server certificates, digital signatures, SSL, Access Control Lists (ACLs), Client Revocation Lists (CRLs), internal authorization systems (LDAP, OCSP, AD, Radius) and how best to use them in support of the application or service.

9.1. Can we use client or server certificates for authentication?
9.2. Will digital signatures be employed?
9.3. Do we need to encrypt all traffic?
9.4. Do we have an internal authority (LDAP, AD, OCSP, or Radius) that we need to authenticate against?
9.5. Will there be parameters that the network can identify to allow only properly formed and /or valid requests?

10. Testing Design and Parameters:
This step should define test plans and methodology to ensure that original design goals are being met.

10.1. In the end, how should the client behave? How should a request traverse the network and to the application?
10.2. What are the target response times?
10.3. Can we pass bad traffic to the application or service?

11. Application and Network Maintenance:

The final step of the service is to leave behind a methodology guide for maintaining the application and network and guidelines for future deployments.

11.1. Document and secure the security model.
11.2. Document and secure the network and configuration design.
11.3. Document the appropriate behavior of the client, the application, and any plans for disaster recovery.
11.4. Are application or service updates anticipated to be frequent and do we want to automate that process?
11.5. Does the application team need access to their production nodes?

These steps and methodology should set an integrated team on the proper course and can serve as a baseline for future projects.

If you have any suggestions for improving this document, please send them to e.giesa@f5.com.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com