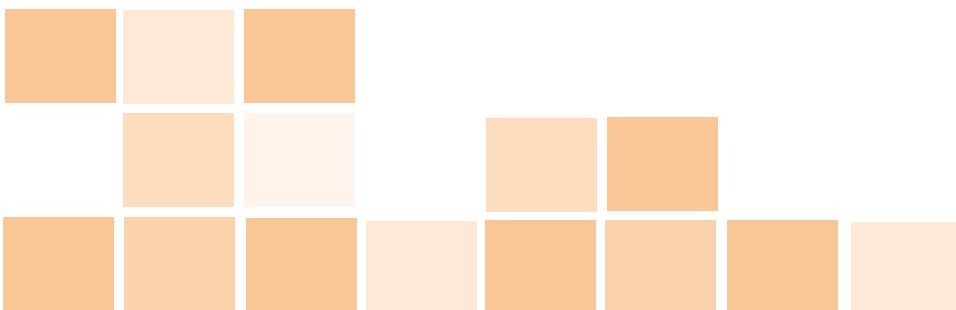




WHITE PAPER

eCommerce and Public Exchanges



Introduction: A public community exchange for eCommerce has several responsibilities to that community. How well the exchange meets those responsibilities will determine, in large part, how successful that exchange will be.

This discussion will focus on the fundamental purpose of any exchange:

To provide for the easy, secure, assured receipt, delivery, and processing of eCommerce documents between trading partners.

In theory, this is a fairly simple task of coming up with a set of standards, implementing those standards using proven technologies, and then publishing those standards for the benefit of all the participants. In practice, however, it is much more complicated. To truly act as an exchange, to truly meet the needs of the members, an exchange must allow for a full communication system regardless of the abilities, technologies, and standards adopted by the members of that community. To further complicate the issue, the exchange must recognize and accommodate the simple fact that not all community members will follow the same path or make the same decisions regarding their standards and technology. It is the exchange that carries the responsibility for ironing

out any differences as seamlessly and easily as possible.

In theory, theory and practice are the same. In practice, they are often quite different.

Nowhere is this statement truer than in eCommerce exchanges. In theory, a set of standards can be defined in the middle and all participants will adopt those standards for the sake of easy communication and the good of the community. In theory, economic and community pressures will force all participants to the same common middle. In practice, the participants will bring their own expectations, their own technologies, and their own eCommerce capabilities to the community. It is the exchange's responsibility to find a way to bring them in to the community with a minimum of directed change for the participant.

eCommerce in Layers: The best illustration of this concept comes from breaking down the exchange functionality into layers. Each layer can then be examined for compatibility problems and solutions. The layers detailed here are not necessarily definitive or groundbreaking, but they do serve to illustrate the problem. They are similar in concept and structure to most network layering exercises, and will be familiar to those that have worked such problems before.

6	Data	We must all use data in the same fashion with the same set of meanings. Whose product designations will we use? Whose abbreviations?	Product numbers, designations for units of measure, numbering schemes, etc. Is a case denoted by 'cs' or 'case' or some other method?
5	Content	We must all agree on common verbiage for the things we talk about. More than just the same language, we must now agree on the same information that we will share.	Field definitions, field usages, etc. Does 'Price' mean net after discounts or gross before discounts? If a product number is used, is it the seller's number or the buyer's number?
4	Topics	Conversations are useful only if both parties understand the topics of those conversations. Our respective systems can work only if we have an agreed upon set of topics.	Purchase Orders, Quotes, Changes to Orders, Acknowledgements, etc. Topics can be document based ('this is a Purchase Order') or message based ('Please Acknowledge this Order as Received').
3	Format	We must agree on a common language, understood by both of us, in which to hold this conversation.	The actual format doesn't matter. Agreement and common definition does. XML, EDI, standardized flat file - all are formats for our communications.
2	Transportation	There must be some method by which we hold this conversation. Two people talk in person by vibrations in the air. They talk over the phone via electrical pulses.	The internet has become the accepted transportation method, although private networks and VANs (Value Added Networks) are still very popular.
1	Connection	Somehow the people holding a conversation must form a connection. Sometimes it is a shared connection (physical proximity), sometimes the connection is through a network (we both plug our phones into the wall, but do not directly connect to each other).	There is a multitude of ways to connect physically to the Internet and through that to an exchange. Generally speaking, eCommerce connections are not one-to-one, but rather individual connections to a centralized exchange.

There are three unique characteristics to such a layered model:

1. Each layer must exist and must be accounted for in every eCommerce environment. There is no skipping a layer, no rearranging of the layers, and no ability to substitute for any given layer. As such, any system implementing an eCommerce solution must find some way to address and accommodate each and every layer.
2. Communication (and consequently eCommerce) takes place vertically up and down the layers. Parties initiating an eCommerce conversation will start at the top with layer six and move down through layer one to complete their part of the conversation. The receiving party will then start at layer one and work up through layer six to fully receive and comprehend the conversation. Responses will, of course, follow the same path in the other direction.
3. Standardization, or at least accommodation of a standard, must exist at each layer between any two parties that wish to hold a conversation. It is not enough to have an ability to move information vertically, that information must be understood horizontally layer by layer.

It is the responsibility of the exchange to accommodate both the vertical usage of the conversation model and the horizontal interpretation of the various layers. In other words, the exchange performs the vertical act of communication and the horizontal practice of understood communications. The illustration below shows these responsibilities.

In this example, the Sending Party has different data values, content rules, topics, format, transportation, and connection methods from the Receiving Party. In other words, this is the worst-case scenario. The vertical column on the left makes perfect sense, as does the corresponding column on the right. Each party perfectly understands their values and methods at each layer, and each expects to not deviate from their well-proven solution. It falls to the exchange, then, to provide for any transformation, translation, or methodology integration at each and every layer. A slightly different solution is necessary at each layer, but each layer will require something from the exchange. Note that at some layers (connection, for example), the choices are fairly limited and there is little need to build new accommodations for every single community member. At other layers (data, for example), the mapping system will require a very rich number of possibilities, with unique maps between almost every community pairing.

A Comparison to Human Interaction: All of this seems terribly complicated. Wouldn't it be easier to simply declare a standard and move on? It would definitely be easier; however, it would also be impractical. Every conversation of every type includes all of the above layers. It isn't always obvious to the casual observer, but the participants in that conversation have negotiated any differences at any level in order to increase the potential of a successful dialog. For example, if two people are talking directly, they have established that the speaker is talking loud enough for the listener to hear the vital parts of the conversation. If this is not true, the listener will interrupt with feedback

Layer	Sending Party		Exchange	Receiving Party	
	Name	Example		Example	Name
6	Data	Product 123	←Data Mapping→	Product XYZ	Data
5	Content	Discount	←Definition Mapping→	Allowance	Content
4	Topics	Purchase Order	←Business Rules→	Order	Topics
3	Format	XML	←Translation→	Flat File	Format
2	Transportation	Internet	←Methods→	VAN	Format
1	Connection	JMS	←Technology→	FTP	Connection

to indicate that the speaker should speak louder. It works every time. But it works because human beings have several advantages not available to computers. First, they have been carrying on conversations for thousands of years and have adapted to most of the known situations in such conversations. Computers have a much shorter history and a much more limited ability to quickly and automatically adapt to the situation. Second, most human dialogs are very interactive, with lots of opportunities for feedback and adjustments appropriate to that feedback. If a listener doesn't understand a given word or a given context, they will stop the conversation and force an adjustment. Or they will pick up the meaning or context from other parts of the conversation. Computer capabilities in these areas are very limited.

With all of these limitations, we have to give computer systems some other advantages to help promote successful conversations. So we give them interpreters. These interpreters help to ensure successful communication between two participants that have either limited feedback loop capabilities or no ability to adapt automatically or manually to differences in communication elements. These interpreters are not unlike interpreters for human conversations. Think of the United Nations. It would be useless to simply declare a standard and expect everyone to automatically adapt. It would be frustrating to simply ignore the differences and assume all of the participants would eventually adjust to everything being said and heard. So they employ interpreters (at several layers, by the way) to provide for meaningful conversations between a large audience of participants. Modern computer exchanges must serve the same purpose.

The Real Reason Behind the Layered Approach:

As suggested above, each layer of the conversation grid serves a slightly different purpose in the conversation. Differences between community members can and will exist at every layer. Taken as a whole, it would seem almost impossible to bring any kind of solution to this electronic Tower of Babel. But broken into layers, it is obvious that a series of tools can be produced to solve the problems one layer at a time. If an interpretation map can be produced between any two participants for a given layer, then the communication between layers and any differences at those layers becomes irrelevant. To

put it another way, an interpreter that speaks French and English can do their job person to person, phone to person, or even e-mail to person! Communication incompatibilities must be addressed at every layer, but not for all layers at once. This approach of separation makes the problem solvable.

The layered approach also points out that a successful exchange will include many tools, technologies, and applications. A data mapping application is good for one layer, but of no use at another layer. A good architect will plan for a transformation mechanism at each layer, but will not insist on the same solution at every layer.

Many-to-Many or Many-to-One-to-Many: There are two possible solutions for the enterprising exchange architect. The first, tried way too often in the world of eCommerce, is to map every single sender to every single receiver. This works great with only two participants, not so great with 20 participants, and not at all with a hundred or a thousand. Called a 'many-to-many' solution, this shortsighted approach often leads to total breakdown as the theoretical number of mappings and combinations quickly climbs to the astronomical. Hopefully, long before that point, the successful architect reverts to a common middle ground - literally.

Sender	Exchange	Receiver
Standard 1		Standard 4
Standard 2		Standard 5
Definition 3	Translated to Internal Standard	Definition 6
Definition 4		Definition 7
Method 5		Method 8
Method 6		Method 9

This technique works at every layer with very few exceptions. On occasion, you will find a data mapping exercise that is truly unique to only two participants and irrelevant to the exchange and any other participants. But those situations are actually more rare than you might think when you open up the conversations to close examination.

Conclusion: The ideal exchange looks something like this: First, it provides for the widest possible choices at each layer. These might be technology choices, application choices, data or format choices, or approach choices. Keep in mind that while 'anything goes' is the goal, it is almost impossible to achieve. Even the U.N. would struggle trying to provide for every given dialect in every possible combination. But that's the goal! Second, the exchange allows for the mix-and-match of those choices in constructing a vertical conversation column for each participant. Want to use flat-files delivered over the internet through FTP? That's fine. Want to provide for screen-scraping of report print-outs through a telnet connection? We can do that. Third, the exchange features a translation system both inbound and outbound that accommodates all of the

community members and all of their choices. These layer tools are distinct and single-purpose, which allows for the combinations just spoken of. And finally, the ideal exchange is built in components so that new methods and transformations can be snapped in easily without disturbing other layers or other participants.

The ideal exchange does all of this hard work for one reason; to accommodate the widest possible number of participants in the community with the fewest imposed changes on the members of that community. In other words, the ideal exchange includes all of these capabilities to ensure its own survival. An exchange that imposes standards instead of accommodating them is automatically limiting its effectiveness, its growth, and its chances of commercial success.

