FIA Investigator Meeting

April, 2012, Fort Collins, CO

Industry structure and business models: Issues with real-world deployment

Meeting summary

Prepared by David Clark

Version 1.0 of September 4, 2012

This meeting had as its theme the industrial/commercial implications of our various FIA proposals. For each proposal, we asked what is an industry/organizational structure such that all the actors are motivated and incentivized to play the role that the architecture defines for them? Is the proposed design viable as a deployed system in the real world, with policy and economic constraints?

This framing of the problem emphasizes the economic aspects of real-world deployment, but other issues came up as well—issues having to do with other aspects of real-world applicability, such as privacy, support (or not) for government intervention, and the like. Users and governments are examples of other actors who have to be incentivized to work within the constraints defined by the proposed architecture. In general, any specific proposal is going to induce an *ecosystem* of stakeholders, and our goal is to understand that ecosystem and its viability.

The discussion at this meeting had a strong technical component as well as economic and other issues of real-world applicability. Since the industry structure implied by a proposal is a derivative of its technical design, the discussion of each system began with the relevant aspects of its technical design.

To help refine the focus of this meeting, we sent out a list of 7 discussion questions before the meeting. The goal was to use these questions to shape the meeting.

The discussion questions

Who are the service providers in your architecture, and what is the resulting provider ecosystem? (Some of the FIA architectures seem to presume a provider ecosystem similar to today: a connected set of packet forwarders. Some presume other services related to carriage, such as storage providers.)

- What is the incentive of each of these actors to enter into their line of business? Where would your architecture require payments among actors to sustain viability?

Options for control: which actors can influence the behavior of a transfer?

- Does your architecture provide user control over aspects of service selection: routes, service qualities, or providers of support service (e.g. like DNS in today's Internet)?
- To what extent does your architecture support or resist the goals of those who wish to control access to classes of information (e.g. governments, rights-holders). How does this position influence the balance of power in your network, and its viability? Which actors have the ability (or perhaps the *easy* ability) to block communication among willing end-points?
- IP addresses accidentally turned out to be scarce resources, for no good reason. What features of your architecture might turn out to be "scarce resources" or resources over which some potentially powerful actor could exercise control?
- Do you have hierarchies with single points of control at the root? Is there information you share with partners that has to be signed by a trusted third party?
- Are there policies that you have explicitly embedded in your design?

What is the range of services that the system provides to the higher layers?

- Compared to today's Internet, would you expect the same sort of commercial entities at the higher layers?
- For example, (especially in the context of those architectures that emphasize information retrieval), would you imagine that there would be CDNs operating on top of your architecture?
- Does your architecture provide an API that defines the service interface of your system?

Interfaces among providers

- What types of information is expected to be exchanged between providers? This goes beyond packet forwarding to include:
    - Routing information
    - Naming information (e.g. DNS zone transfers)
- An interconnection agreement between providers in today's Internet may have Service Level Requirements, or specify aspects of routing policies (cold potato, hot potato). What would you expect to find in inter-provider agreements in your architecture?
- To what extent do services provided to higher levels (see above) require negotiation or cooperation among the various actors that make up the overall network?
- What mechanisms does your architecture provide for negotiation among service providers?
- What range-of-functions are supported by the protocols and mechanisms that hook them together?
- Operators are sometimes worried about all getting together to solve operational issues. It is hard to do and looks like anti-trust. What are the "top five" aspects of your architecture that require operational coordination?

Market forces and regulation

- To what extent does your proposal facilitate or limit the use of competition as a discipline on the market?
- If regulation were proposed to require some sort of non-discriminatory access or "network neutrality", what might that mean in your design? Where might forms of discriminatory service emerge?

Evolvability

- How does your architecture allow innovation and the migration to new mechanisms?
- Which sorts of evolution seem to require global coordination, like the migration to IPv6 today?

Trust, isolation and availability

- What sorts of trust assumptions does your design make about the various actors that make up the ecosystem?
- Does your architecture provide means for instrumentation or data-gathering? What sorts of data? Internal structure of the network, usage, routes, outages, etc?
- To what extent does your architecture include tools to detect that actors are not functioning properly? Which actors have access to these tools?
- How do your options for control allow different actors to respond to actors that are not trustworthy or mis-functioning?
- Availability often implies "extra" or "diverse" resources. Does your architecture depend on resources that are otherwise under-utilized to achieve high-availability. Is economics a barrier to a high-availability network? Both within a region and across regions, does your design allow the operator to trade off explicitly between cost and availability/resilience?

Project summaries

The meeting opened with short descriptions of the five projects. Four of these—NDN, Nebula, MobilityFirst and XIA, have been described before. See the summary of the May 2011 meeting. A new project, ChoiceNet, was presented for the first time at this meeting.

ChoiceNet

In contrast to the other FIA projects, which describe a specific forwarding mechanism (e.g. the data plane), ChoiceNet is focused at a higher level: the control plane and what they call the economy plane. The assumption is that the data plane (which might be implemented using one of the other FIA proposals such as Nebula) will provide alternatives or choices among services: for example IPv4, IPv6, different paths with different qualities, or other services in the network. For

example, a user might choose to pay more to get a higher quality movie. The goal is to make these options explicit and allow the user to pick among them.

The assumption is that the network will have a connection setup phase, and the user will express his choices at that time. The setup phase is implemented in the control plane, where component services can be selected, or composed to make new services. The result will be implemented in the data plane.

A way to conceive of the economy plane is to think of an app store for services. Different service offerings would be advertised, there would be rating systems, and so on. The user would make simple choices, which would translate into actual services by the composition of elements in the control plane.

Introspection or verification is an important part of a control plane. Did the user get what he paid for? ChoiceNet includes components that measure what is going on to verify what has been provided. More specifically, an overall service may be composed of many parts. They all get a share of money, but should also get the correct share of blame for failure. So ChoiceNet will provide some external verification as part of any service. Service proofs and payment verification are exchanged between data and control plane.

The term "user" in this discussion might be an actual person, or a software agent, or expert (human agency) like a sysadmin setting up service for user.

Over the course of the meeting, there were a number of discussions about the ChoiceNet approach. A recurring question had to do with whether the users can realistically make choices, whether the offered services (in the service app store) will be well-enough specified that the user will not be misled, and so on.

Use cases

The participants at this meeting from the Values in Design Council suggested an alternate way to compare these architectures. They proposed that we use specific use cases or scenarios to better understand the mechanisms of the different proposals: how would they deal with the issues in each case. They proposed four cases, as follows:

1) **Reporting a protest.** Imagine that there is a mass protest or demonstration going on, perhaps with repressive action from the authorities. One user records this event as it is happening and uploads it, through some Internet architecture. The video of the conflict is streamed in real time to an audience that grows from a few to many thousands. The government in question wants to repress the sending, and for its citizens prevent it being viewed. This example is interesting because it is important but not commercial. It brings out the adversarial relationship among actors, with the adversary having reasonable resources.

2) **Facilitating broadband deployment and user choice.** In the U.S. today we see two facilities providers in some places (e.g. cable and telephone) and only one (or none) in others. Does your architecture provide an improved opportunity for a third provider to enter into competitive markets, or for a single provider to enter markets (e.g. sparse rural areas) that are not economical today? For example, might your architecture make it easier for small, independent wireless providers to enter the market. Can your architecture encourage this sort of innovation?
3) **Gambling.** (A detailed description of the scenario can be found at the end of this report.) Online gambling is sometimes legal, sometimes not, depending on the jurisdiction. It is provided by very well-funded and motivated actors to customers who very much want to participate. On the one hand, users need strong assurances of correct, timely operation and trustworthy payoffs. On the other hand, jurisdictions that ban online gambling try to block, prevent, or suppress the activity, perhaps by disrupting the banking and payment schemes. The providers respond with very complex, cross-jurisdiction hosting architectures. Compared to today, does your architectures change the balance one way or another?
4) **Just and fair treatment in times of disaster.** In a disaster like Katrina, computing and communication resources are disrupted. At the same time, the needs for critical information may be even more important. Imagine an elderly person with poor computer skills, who needs access to medical records. Does your architecture change her ability to get to her records?

The questions

The bulk of the meeting was a discussion of the seven questions in turn.

Who are the service providers in your architecture, and what is the resulting provider ecosystem? What is the incentive of each of these actors to enter into their line of business?

ISPs: All these schemes presume that there will be private sector actors, similar to the ISPs of today, that provision, control and operate regions of the network. In general, these architectures give them a larger role in the operation of the network.

- NDN: they are responsible for the dynamic caching of packets of data, validating the legitimacy of the data, and so on.
- XIA: they provide a range of services (tied to types of XIDs) that can include content caching, multicast, anycast to replicas of service or content, and so on.
- Nebula: they provide a validation that packets have followed the path that was generated by the data plane.
- MobilityFirst: like XIA, ISPs provide a range of services; they also host third party computing services on their infrastructure and provide mobility-specific services such as short-term caching, redirection and the like.

Collectively, they implement the core function of binding name to location, the GNRS.

- ChoiceNet: the data plane is not specified in ChoiceNet, but it must provide a set of interfaces to the control plane, through which the data plane can be configured to deliver services. Enhanced services, and the ability for the user to select them, is the central point of ChoiceNet

Name services: All these schemes require some sort of translation or binding service that maps from high-level names to lower level identifiers. This service would resemble to some degree the DNS of today, but all the proposals that discuss this in detail propose that there should be multiple, competing naming services, to avoid a single root of trust. This might lead to specialty name services that could be offered by smaller, independent providers for certain classes of names. Providing certified names might be a valid business, in that one might pay to have a certified name.

Other services: Several of these schemes describe third-party "in-network" service providers that can be configured into the forwarding data path. In particular, Nebula and ChoiceNet use the power of their control plane to control the integration of such services into the forwarding path. This would create opportunities for both ISPs and third parties to offer new services.

ChoiceNet, because it is focused on the higher layers in the architecture, identified a number of new classes of actors, including service authors (who invent a new service), service hosts (who provide the infrastructure for the new service), integrators that build new services out of existing offerings, payment/broker services and verification services (which provide assurance that the requested service was actually provided).

The role of the ISP: This question resulted in an interesting discussion of the role of the ISP. NDN takes a distinctive point of view in this respect. In most architectures, including today's Internet and several of the FIA proposals, an ISP provides (part of) a connection to a named endpoint (e.g. a host or a service). In NDN, the ISP satisfies a request for a set of bits. Abstractly, NDN brings a focus on the outcome (the bits) and not the process (the connection). The user has a bilateral relationship with his local ISP, and the ISP is free to satisfy this as it sees fit—it is a local arrangement on both sides. The ISP can establish a range of agreements with other providers in order to fulfill the agreement, but these are not a part of the service specification. In contrast to schemes such as ChoiceNet or Nebula, which give the user (or software acting on his behalf) the ability to select and compose a rich set of services, NDN takes the position that the any explicit service selection is done by the ISP, and the user makes an implicit service selection by using the ISP that provides the best outcomes.

This view implies that the user should be given the option of diverse multi-homing, which thus becomes part of what NDN needs to fulfill this aspect of its approach. Unless NDN provides some sort of overlay or "virtual multi-homing" capability, this

view would seem to imply the need for "facilities-based" alternatives. However, it is not clear that there are features in NDN that would make it easier for competitors to enter the access market, as was described in the **Facilitating broadband deployment and user choice** use case.

To some in the meeting, the NDN focus on "bits as the outcome" seemed un-natural. The example of banking was raised—in that case it seems that the user and the bank are each concerned with knowing that they reached the other. In the banking case, the desired outcome can still be modeled as whether the receivers are getting the correct bits. In this case, most of the bits will not have been cached, but the point of view that looks at the outcome rather than "securing the connection" is still valid. There is a duality between looking at the process or the outcome. The claim of NDN is that you get more mileage and decoupling if you look at outcome and leave the process unbound by the architecture.

In contrast to the NDN approach of a single service provided by the ISPs, XIA has a very general architecture in which different sorts of service options can be embedded, to be explicitly selected by the user (via, for example, the class of XID). The actual character of a deployed XIA network will depend on which services are actually standardized and deployed. That generality is part of their approach to evolution, and as well the basis by which different actor ecosystems can emerge.

The role of advertising: The role of the advertiser was brought up in the discussion of the relevant actors. Advertisers matter because they are an important part of the payment ecosystem. Advertisers have requirements that may fit more or less naturally into these architectures. Advertisers want to place ads into content that are customized to the receiver, so they need to know who the receiver is, and they need accounting assurance that the ad was actually placed. The simple story of "retrieving a page" does not capture the complexity of this reality. The discussion suggested another possible use case:

Complex web objects use case: a web page is made up of many parts, a outer page, embedded parts provided by the creator (e.g. images), embedded parts provided by third parties (e.g. advertisements), analytics, cookies and trackers, and so on. Many of these will depend on the end-to-end tracking of the identity of the receiver. How do these different architectures deal with this more realistic download case? Could your architecture allow or facilitate different patterns of money flows (e.g. to ISPs or content routers) that would be beneficial to the various parties?

Incentives: One part of this question was whether all of the actors would have an incentive to take on the role that the architecture defines for them. The various presentations did not consistently address this question. Several proposals invoked the power of user choice as a discipline that would create a market, e.g. in ChoiceNet or Nebula's NVENT with their explicit service selection, or with the implicit service selection implied by connection selection in NDN. However, there was skepticism that this choice could be exercised effectively by normal users, and also skepticism

that the major actors would be motivated to offer these choices. This is discussed further below.

Options for control: which actors can influence the behavior of a transfer?

No architecture can sidestep the basic consideration that a given ISP has ultimate control over the traffic going through it. The different proposals try to counter this by empowering the user with choices as to which actors are involved in the transfer. However, choice takes very different forms in the different proposals.

Nebula: Nebula provides a powerful control plane, NVENT, in which the user can exercise explicit choice over the actors that carry out the intended operation. This is encoded into a source route used by the data plane, NDP, which provides assurance that the packet actually took the path through the actors selected by the user.

ChoiceNet: ChoiceNet is very similar to NVENT in this respect—it is a control plane where paths and services can be negotiated. It does not specify how the data plane enforces the path, but assumes the capability exists. The motivation of ChoiceNet is to allow smaller players to enter the market and offer services, leading to an ecosystem more like the space of Web applications than the current concentrated carrier world. It is this diversity that will provide user choice and discipline the providers not to exercise unwelcome control.

NDN: NDN, in contrast, gives the user control by implicit means. There is no per-flow control plane in NDN. NDN assumes that the end-user (the requestor of information) is multi-homed, and will try different options to send his requests (interests). If an ISP cannot provide service, users will not continue to select it. This bi-lateral relationship cascades between all the ISPs in the net: each ISP is responsible for finding an upstream ISP that can satisfy interests.

XIA and MobilityFirst: Both of these proposals take a somewhat traditional approach to control, similar to the Internet of today, in which the ISPs compute routes and user traffic follows those routes. However, MF, with its emphasis on the access network, allows the user a high degree of multi-homing and multi-path, which will allow the user to pick paths that work. Both MF and XIA (through the SCION technology) provide a form of source routing that can be used to bypass points of disruption.

Choice and "openness": This emphasis on choice, and in particular the approach of explicit service composition, triggered some commentary from the Values in Design group. The design of the Internet, which has very few options for explicit control, provides a default platform with a baseline expectation of service. All traffic receives this treatment, which led to the characterization of the Internet as "open" or "neutral". In a world where service is explicitly negotiated, this default expectation may not exist. In many respects, this world might seem to give the user more choice, but it will also give the providers more options to discriminate. This world may in fact not be "open" in the same way as the Internet of today.

This observation ties back to the discussion of incentives, in the previous section. Providers will offer choices only if they are motivated to do so. The obvious motivation is payment, which (as ChoiceNet makes explicit) ties choice to the economics of the service providers. In fact, choice, because it transfers more control over resource usage from the ISP to the user, may drive overall costs up.

In particular, as we contemplate the emerging power of commercial content, and the desire of different actors to benefit financially from the carriage of this "valuable" content, it is not clear what the motivation will be to offer the end-user choices in its delivery. What we see today in the Internet is intense concentration of power. It is not clear that any of the FIA proposals do anything to reverse this trend. It would be nice if every proposal make clear the extent to which (and in what ways) they hope to shape the market.

Who is the user? This discussion was an echo of the one at the previous FIA meeting, which discussed choice as a component of a trustworthy network. The actual end-user will not be sophisticated, and cannot be expected to make detailed technical choices. He will need to invoke some agent, either software or human expert, to make choices on his behalf. But in practice, users will often pick the default option, so the specification of what that default is will greatly shape the user experience, and the balance of power in the system. Will the default be similar to what today we call the "best effort open Internet", or something else?

ChoiceNet: This design, as a part of its "economy layer", proposes the concept of an "app store" for services. The app store would offer high-level services for the user to select; the provider of the "app" would then make the lower-level decisions to compose this service out of the parts provided by the data plane.

Scenario discussion—the dissident reporter and the government: as a way to get at the question of control, the group considered one of the use cases proposed earlier, which was the person making and disseminating a recording of a mass protest, which the government attempts to block.

MobilityFirst: The lack of a control plane makes it harder for third parties such as governments to intervene. The sender can pick lots of paths into the network, and can use short-term GUIDs for both source and destination. In one approach, the content would be sent ("uploaded") to an intermediate server outside the jurisdiction, which divides the problem into two parts, blocking the upload and blocking the download. In another approach, local observers would attach to a multicast stream of the content, which would tie the GUIDs of the senders and the receivers. However, to map the receiver GUID to a specific host, one would have to have access to the low-level route maintenance data. One could use a "TOR-like" rewriting of GUIDs to help hide.

Nebula: the ability of governments to block or trace communication depends on the design of the NVENT control plane. If the user has choices about his initial entry-point into the network, and the negotiation in the control plane reveals his

intentions only to the requested actors, the user may have some protections. If the user only has one option for his access to the network, and the government can observe the NVENT negotiation, there is not much protection from surveillance and control.

ChoiceNet: if ChoiceNet is successful in creating an ecosystem that allows lots of small actors to enter the market, the options for control are diffused. Lightweight service formation by groups of protestors might be a powerful building block of a robust capability.

XIA: XIDs (e.g. content ids) can be transitory if the rebinding from one id to the next can be managed. The issue is to manage rebinding in such a way that the government adversary cannot track it, but the intended recipients can.

NDN: Because NDN is a "pull" architecture, the analysis is very different. Whether the video is being "uploaded" to a server, or directly watched by other viewers, the data leaves the machine where it is being recorded only because interests are being received. If many viewers send interests that arrive via different machines, the data will leave via those different machines, which will allow many transient copies to be created. If a government is trying to trace back the data packets to the origin, this will be very difficult. The interest packets do not have the identifier of a machine in them, just the identifier of the data. If a government agent can get access to one of the intermediate machines, all that can be discovered is stored interests, cached data, and the current information that is guiding the strategy module, which (if the machines in question is part of a mesh radio network) will just be an indication that if a similar interest is received in the future, it is appropriate to broadcast it onward. Even if the agent is a single hop from the origin, he cannot tell if the next hop is the origin or just another relay node.

The power of user choice:

The preference for user choice is based on the assumption that user choice is a more powerful discipline on unwelcome control than the "we are all in the same boat" of the open Internet. This approach begs the question as to whether choice will actually be made available in the running network, and whether choice will be an effective tool.

We have no way to quantify the dimensions of choice and control, so it is difficult to argue that one scheme is better than another. In particular, with respect to the scenario concerning the dissident recording, without a flexible threat model of what the adversary can do, it is hard to have a meaningful discussion. None of the architectures seem to shift the balance of power in a qualitative way.

What is the range of services that the system provides to the higher layers?

The different projects have very different views on this question.

Nebula and ChoiceNet: these designs assume that service building blocks in the network can be composed to present a rich selection of end-to-end services to the applications.

XIA and MF: these designs provide a small number of service classes, corresponding to different classes of IDs—for example content, services and hosts. Each of these classes would correspond to a forwarding behavior in each router. MobilityFirst also allows for additional functions to be installed on routers in the path. MF does not support per flow QoS.

NDN: this design implements a single, general content retrieval service. It allows for variation in service quality (e.g. QoS) using a field in the packet similar to the IP header of today.

One way to understand these distinctions is that if the set of anticipated service classes is limited and specified (as with XIA) the relationship between the provider behavior (or a router behavior) and the resulting end-to-end service can be worked out as part of the specification of the service. On the other hand, if the set of anticipated services is open-ended (as the example of the HIPAA-compliant path used by Nebula, or a path that avoids a particular region of the world), the composition of the service from component parts must include end-point control over the path, and a more complex and sophisticated composition algorithm, which implies a separate control plane.

What does a router do: One possible way to better understand these different schemes is to compare the range of end-to-end services to the specification, for each scheme, of what a provider must do.

XIA: the basis requirement for a router is that it implement the extensibility architecture—the ability to parse the DAG in the header. In addition, it must implement some of the expected service classes, e.g. it must deal with host IDs, service IDs, etc. But the DAG and the resulting fallback mechanism means that not all routers must do the same thing. The composition of the per-router or per-provider function into the end-to-end service is *implicit*; it is done by the route computation algorithm. (But see the discussion below on payment as an explicit linkage among providers.)

MF: routers in this scheme must support the defined service classes, as with XIA. There is no fallback scheme, so more homogeneity is required of the routers. Routers perform hybrid routing, using either the GUID or the NA. Again, the composition of the per-provider function into the end-to-end service is implicitly done by the routing algorithm. In addition, MF routers must participate in the implementation of the GNRS that maps GUIDs to network IDs.

NDN: NDN routers are very different in concept from routers in other schemes. They receive *interest* packets, record a per-packet entry for each such *interest*, implement some *strategy* to forward the *interest*, and if a *data* packet is received in return, they use the stored per-packet information to send it on. There is no specified routing

protocol—the composition of the per-router behavior into the end-to-end service is based on the local decision at each router as to how to forward the *interest* if there is not a locally cached copy of the *data.*

Nebula: In this scheme, routers (and other service elements) are assumed to implement a wide variety of functions, including traditional forwarding, computation, and so on. These functions must be well-specified, but the architecture does not determine these specifications. The composition of these functions into an end-to-end service is performed *explicitly* by the NVENT control layer. The Nebula router must comply with the forwarding instructions computed by NVENT: if a packet is properly received along the path computed by NVENT and specified in the packet header, it must process and forward it according to the path.

ChoiceNet: this system does not specify a data plane, but is very similar to NVENT of Nebula, in that it provides a control plane that can compose lower level service elements into an end-to-end service.

Verifying the service: Since a service is built up out of the local functions performed by the different providers, it would seem important to have some way to verify that the expected service has actually been delivered. Different designs take different approaches.

XIA and MF: these schemes resemble the current Internet in that they do not explicitly verify the service, but leave it to the end nodes to detect that something is wrong.

Nebula: the Nebula Data Plane, based on the Icing scheme, provides explicit verification that the packet has followed the path computed by the NVENT layer. It does not provide means to verify that other aspects of the intended service have been delivered, e.g. if the various components did what they committed to do. If the end-nodes are not satisfied with the service as delivered, and can isolate the source of the impairment, they can use NVENT to avoid that provider.

ChoiceNet: this scheme makes verification a core part of the architecture. They assume that there will be specialized providers that implement verification services, which will include monitoring elements positioned between providers. Of course, verification is only meaningful in the context of a particular service, so verification is a complex research problem.

NDN: verification of service is central to this design, but because of NDN's different overall approach, verification takes a different form. Since routers keep per-packet state, and since packet flows in NDN are bi-directional (a data can only flow back along a path defined by stored interests), every router can monitor the timeliness and reliability with which a forwarded interest is matched with a returning data. This means that every router can continuously evaluate how its "upstream" providers are doing, and pick among them accordingly. However, there is no explicit specification of the service parameters, as there might be with Nebula or ChoiceNet.

There was an interesting discussion of verification in the context of highly reliable services. For services that are designed to be very reliable, failures essentially never occur, so one cannot verify the level of reliability by counting failures. This fact can lead to a problem of "free riding", in which providers do not actually implement the reliability measures they committed to. There can be a number of possible market failures around the construction of cross-provider reliability, including the problem that the customers may have no way of confirming that they are getting the service they paid for.

Does the architecture specify an API? Application designers need some well-specified interface by which to invoke network services. However, none of the proposed architectures include an API as a part of their specification. This observation triggered an interesting discussion. In general, the different architectures share the view that what really matters is the treatment given to packets inside the network. There can be more than one API that gives access to that behavior (or a subset of that behavior). So an API should not be specified as part of the architecture, but "wrapped around" the architecture.

On the other hand, from the perspective of formal methods, it is the interface that defines the module, not what goes on inside. The interface, once specified, is hard to change, and may end up defining (in engineering terms) what the system does.

Nebula: This system is using the Serval system to avoid the potential ossification that can arise from an API.

In fact, many aspects of the service provided by the network (or composed out of service elements by a control plane) may not change the API of the service. Picking a path that avoids a region of the world, or a "HIPAA-compliant" path, or a QoS that reduces latency, does not necessarily change the API to the network. Rather, they imply a very flexible and expressive interface to the control plane. Designs that do not have a control plane, such as NDN, must use the data plane to trigger any service variants, which implies that the API must provide access to these services, for example by setting TOS bits in the header.

Services and the application marketplace: The VID group commented that our discussion of services, and the relationship between the proposed architectures and the higher layers, seemed to have a passive sense. Echoing the theme that design is policy, they asked whether we should be trying to shape the higher levels through the design of the architecture. We should not be reluctant to think about the uses of our architectures, and in particular the commercial users of the architecture.

The market is not good at solving every kind of problem. From their point of view, architecture is an alternative to regulation to shape those things the market may not do well, such as enforce network neutrality if that is desired. As an example of a possible architectural value, will the caching approach of NDN favor the efficient distribution of popular content in a way that erects barriers to less popular content? As a counter to this point of view, it was observed that the market has a lot of

feedback around policy, and the degree to which value can be baked into an architecture depends on the degree to which removing it actually breaks the system. Almost all of the architectural assumptions of the original Internet have now been violated, but the system somehow still works. The evolution of the Internet (and its architectural assumptions) just reflects the emerging preference of the powerful users of the architecture.

Interfaces among providers

One way to organize the answer to this question is to note that information is exchanged among providers at different levels. Information such as routing is exchanged as a part of a well-defined control protocol. Many of the designs assume a somewhat traditional interdomain routing protocol, with the exception of NDN, which is distinctive because it routes on names, and does not have machine addresses that can be the basis of a traditional routing protocol. Designs with a rich control plane can exchange (or expose) a lot more information at that level. Nebula and ChoiceNet allow very complex interaction among providers, but these are not implemented as protocols in the data plane.

Economics: The discussion quickly turned to the point that in several of these designs, there is a richer set of "interfaces" for the exchange of money than in the current Internet. These "architectures of economics" are a part of the incentive structure that is intended to make the architectures viable. With good designs, the creation of economic interfaces will not just lead to the redistribution of money among the actors, but will lead to a "bigger pie to divide": the creation of new services that one or another actor will be willing to pay for.

XIA: The designers gave a lot of thought as to whether the implementation of the defined services required negotiation among the providers, or whether they could be composed out of the local router function operating independently in each region. They concluded that getting the flow of money right was a central issue—what is needed is some sort of broker or "economics overlay" to avoid the necessity of having a NxM negotiation problem between all the regions and all the customers. CDNs do this today—they negotiate with content providers on the one hand and ISPs on the other hand.

NDN: CDNs could play a similar role in NDN. A CDN overlay on top of NDN could always replicate the content to provide multiple origins for retrievals, but since any router can be a cache, the CDN could also negotiate with the provider of the router to store the data, and receive payment in return. This might be cost-neutral for the CDN (pay the ISP rather than deploy more caches) and provide a new source of revenue to the ISP.

ChoiceNet: The control and economy planes of ChoiceNet are exactly these brokers that allow money to flow into different parts of the forwarding chain without having to flow in from the edge along the data path. As part of this economic ecosystem, ChoiceNet allows a provider to advertise service offerings, both to end-users and to other providers. Negotiation in the control plane, which lead to per-flow path

establishment based on this information, might actually render protocols like BGP unnecessary.

Design by specification vs. design by markets

It was observed that in the "old PSTN", interfaces among providers were specified in detail in many documents from the ITU, which described, among other things, what could be measured at what points, and the range of allowable values. The maintenance of service quality was based on these specifications. In contrast, the current Internet (and these new proposals) do not stress formal and universal interface specifications. Rather, they have assumed bilateral inter-provider agreements driven by market needs. However, in both cases, the deployment of cross-provider services seems to call for some form of verification—without verification, the provision of cross-provider services may founder on squabbles over performance. For this reason, as discussed above, several of the FIA proposals include a discussion of verification, either explicit (as in ChoiceNet) or implicit to the data plane operation (as in NDN).

It was noted that in contrast to the world of the ITU, where a single service (telephony) was well-understood, the Internet and its successors will provide a range of services, with a great deal of generality, so it is not obvious what to measure or verify. Some measures of quality may only be visible at the edges. How should these be related to the respective performance of different providers? This is a general issue across the architectures.

Market forces and regulation

The presentations by the projects only gave a partial view into the issues behind this question. A useful way to begin might have been to ask what aspects of each architecture do *not* encourage competition, rather than emphasize their general preference for competition and choice. However, some interesting differences emerged in the discussions.

NDN emphasizes the *implicit* optimization of content delivery, with opportunistic caching and delivery. Given this, without some accounting overlay, there is no way to tell how many times content has been delivered, and to whom. On the other hand, XIA talked about using the XIDs as an explicit basis for accounting for content.

The delivery model of NDN also seems to confound any simple approach to imposing some sort of network neutrality regulation on the network, if it were necessary. Can the provider make arbitrary choices as to what data gets cached? Can it sell its caching service on discriminatory terms?

Nebula: it seemed that there would have to be a competitive market for data centers and cloud, but the discussion seems to talk about the interconnection of cloud as a function within a firm. It is not clear how Nebula builds its core out of competing parts. Is there competition for the paths among the parts of the core?

MF: this proposals hopes to change the competitive landscape in significant ways, most obviously by allowing access ISPs and current cellular providers to compete more equally.

ChoiceNet: this design provides lots of opportunity for competition among service providers of various sorts, including authoring, hosting, verification, and the like.

A way of thinking about the industry structure implied by each of these proposals was to ask how they would shape (or be shaped) by a powerful actor such as Google, which embodies both horizontal integration (e.g. monitoring and aggregation of data across applications) and vertical integration (as they get into the access, handset, and OS lines of business). Every powerful actor wants to "own the customer"; do any of these schemes change that landscape?

The "app store" concept from ChoiceNet may facilitate user choice, but the possible downside is that the provider of an app may not disclose all of its attributes. For example, if an app does user tracking as a side-effect, this may not be disclosed unless forced by regulation (which will always be lagging the market). It is not clear how these sorts of values would be manifested in a pure market economy. ChoiceNet is proposing to use Semantic Web methods to build ontologies of choices and their meanings, and to build tools to use this information to allow more intelligent choice by consumers.

Competition and virtualization: A key question in all of these schemes is the relationship between the provider of the basic services and the owner of the infrastructure. In simple terms: who owns the wires?

Evolvability

A three part taxonomy was proposed as a way to answer this question:

- How do we evolve from the present Internet toward your architecture?
- How does your architecture evolve (or not) to deal with new requirements?
- How can innovation occur *inside* your architecture?

After some discussion, the meeting focused on the second two of these options, and decided that migration from the present was not the critical topic to discuss. The discussion also suggested that perhaps the topic had been mis-named. It might have been called *longevity*: how will your design survive over time. Evolvability is only one answer to this question, as was pointed out in the discussion.

NDN: this system does not emphasize evolution as a basis for longevity. The design philosophy of NDN is that if the system offers a general, flexible base service, and the service meets the needs of a range of applications, then that system can be long-lived, just as the Internet has been, without needing to evolve the architecture itself. If successful, NDN would be general enough to trigger innovation on top of the NDN layer, and innovation in the business relationships between upper layer services and the ISPs.

XIA: in contrast, XIA allows the definition of new classes of XIDs to deal with new service needs, and for the graceful introduction of these classes through the fallback mechanism. So XIA takes the approach of evolving the architecture. It is an unresolved question as to when a new class of XID would be required as part of evolution. When would the alternative of defining a new service with a new SID be sufficient?

MF: this design also has a number of ID types, but does not include a migration scheme like fallback, which would make the introduction of new classes more difficult. This system also allows the deployment of third-party computations on in-network devices, although it is not clear how these would be exploited by an end-user.

Nebula and ChoiceNet: these systems allow for the development of new service elements in the network, and the use of these elements by means of the control plane. So these systems take the approach of allowing innovation inside the architecture. ChoiceNet does not specify a data plane, so it is not clear whether that part of the system would allow innovation or evolution. The high-level description of Nebula would allow alternative data planes other than the one defined by the Icing scheme, but it is not clear that they actually imagine that the data plane of Nebula would be replaced if the system were successful and running. ChoiceNet does not embed specific design decisions (such as the size of an ID), but rather embeds the idea of a marketplace driving the actions of a control plane. In this respect, what ChoiceNet brings is a point of view, just as NDN has its own (very different) point of view about what is fundamental. It was speculated that if ChoiceNet is successful, the interfaces in the control and economy plane (e.g. the building blocks of the marketplace) will be very hard to change. We have little experience in the design of markets, so this issue deserves attention.

All the systems seem to call for multiple name resolution services to map from human-meaningful names to self-certifying IDs. This approach suggests that new services could be introduced over time, in contrast to the situation today where a new DNS could not in practice be introduced, even though it is not a specified part of the Internet architecture.

From a VID perspective, two points. First, the user experience can change for a number of reasons. Many parts of the user experience arise at higher levels, above the architecture. So what would be the driver for evolution? Not the users directly. Perhaps the application designers? This raises the second point, which is that there may be many actors interested in evolving the architecture in different directions.

A very powerful insight was that if the architecture is designed to allow for evolution of the architecture itself, and particularly if this evolution can occur in a incremental and fragmentary way (as, for example, with the fallback scheme of XIA) then actors such as governments may drive evolution in different directions in different jurisdictions, leading to a heterogeneous and perhaps Balnakized future Internet. An architecture that takes a stronger stand in favor of a presumably

general and useful but fixed mechanism, such as NDN, may be less amenable to evolution that fragments it. Do schemes such as XIA need mechanisms to regulate evolution, and is that even possible?

It is important to think about the regulatory context, at least as it is currently framed, because it may act to prevent classes of behavior that look like discrimination. Even QoS is suspect, so the benefits and risks of the rich service architectures of ChoiceNet and Nebula need to be described in terms that regulators can understand.

A very interesting, if negative, comment about evolution was the challenge to each project to think of external events that, if they happened, would derail your proposal. The rapid uptake of smart phones and the global deployment of base stations essentially killed off low earth orbit satellites. Events could be technical, such as a flaw in a public key system, or economic outcomes such as a stagnation of investment. Which of these proposals would survive without a competitive physical infrastructure?

Trust, isolation and availability

The discussion of this basket of questions essentially focused on trust, and revealed to some extent the different sort of trust assumptions that should be considered.

All the designers acknowledged that there might be "bad actors" in their system, or actors with interests that were adverse (without branding one of these actors as "bad"). In some cases, designs seem to depend on trust that actors will not mis-behave, in some cases they depend on verification to detect bad actors, and many of them depend on user choice to select paths that do not contain these bad or adverse actors. Putting that differently, correct operation depends on the ability of the user to select trustworthy actors, and the ability of the design to give the user that degree of choice. In general, the descriptions of the projects did not make clear when the components (e.g. the actors such as ISPs) can protect themselves from each other, and when they depend on trust for correct operation.

One dimension is the range of trust that the system seems to inply among end-points. For example, in XIA and MF, one can make use of end-point identifiers that are essentially anonymous, because they are privately generated. The use of these would seem to imply that in the absence of a higher-level mechanism to exchange credentials, one end-point would have no way to know who the other end-point was. Under what circumstances would users agree to communicate using anonymous IDs, with no knowledge of identity and presumably no basis for trust? It could be that the primary use of dynamic, anonymous IDs would be for unwelcome activities such as spam

MF depends on a global DHT scheme to map GUIDs to network addresses. Since any one router in the scheme might be untrustworthy, they use 5-way redundancy to detect and correct malicious answers. Since the five elements are picked based on a

hash of the GUID, there is no way that five colluding elements can conspire to disrupt the system.

In the original design of the Internet, it was a goal that if two devices were directly linked, they could exchange packets without depending on third-party servers and services. However, making this work is complex in practice, and as the Internet matured and became more of a managed service, it was easier to presume the availability of services such as DHCP, DNS, and the like. Each of these must be trusted to work as specified if machines in the Internet are to communicate. It would be helpful to catalog the extent to which each of the FIA proposal have such services, and thus such dependencies, in their design. One of the design goals of NDN was that it would not need such services for basic operation.

The role of a name resolution system: All of the schemes that describe a data plane exploit some form of self-certifying identifier to allow the ends to confirm that the network-level action has occurred correctly: the host, service or information was the "correct" one. However, these mechanisms achieve their desired objective only if the users can be sure that they have the correct ID for the desired object. For example, if the user has an ID for "the New York Times", then he can tell if that is what he actually got. But he first has to be sure that the ID is actually for the Times, and not a forged variant. This step requires that there be some sort of higher-level name resolution service that can map user-meaningful names to self-certifying IDs in a trustworthy fashion.

All of the proposals assume the existence of such a trusted system. They do not specify it, but just require that it be present. And all of the proposals suggest that there should not be just one name resolution service, but competing services. This avoids (arguably) needing a single root of trust, allows users to avoid services that prove untrustworthy, and allow different name resolution services to operate at different levels of trust to deal with different needs and circumstances.

The "desert island" scenario: This scenario was posed as a way to understand the minimum set of services necessary for different architectures to function, and in particular to understand the difference between the way NDN and the other schemes implements information identifiers.

The schemes except NDN use a hash of the content as their ID for information: the CID of XI, or the content GUIDs of MobilityFirst. NDN takes the different approach of a structured, lower-level name, and an explicit signature of the content and name, signed by the owner/creator. The fact that these lower-level names have structure in NDN allows more functions to be embedded in the data level of the architecture, not left to the naming service on top of the architecture. For example, in NDN a person can construct a name for "today's New York Times, and ask NDN to look for it. If two users are on a desert island, and one happens to have today's New York Times on her computer, NDN would allow the other user to retrieve it using that name. The other schemes would first require a query to a service that would translate the high-level "today's New York Times" to a suitable lower-level self-

certifying ID. Unless the desert island had access to this service, the only way to transfer the information from one computer to another would be for one user to read out the content ID, and the other user to type it in. NDN names, because of their structure, allow a range of actions to be carried out without reference to an external service, such as name-driven routing, and other forms of discovery.

Final comments from the VID participants

Remember that designers are not expected to determine the outcome in a value space, but just to provide affordances to allow the desired set of outcomes. External actors may thwart the desired outcomes, but designers can "tilt the playing field" by use of defaults. And do not be afraid to say that there are certain desirable things that an architecture cannot achieve on its own, and to appeal to other tools such as law and regulation to complement technology. A useful question for each design is what sort of legal and regulatory context would sustain and support their design. Law is not a constant, but a variable in the design equation. Also remember that constraint by law may not lead to the best outcome.

As a form of use case, think about the range of users and applications that will exploit your system, including marginal and unwelcome uses. For example, think about gamblers (and other such marginal users), designers of crypto currency and anonymous transactions, the pirate content community (including those who modify the content, as by adding sub-titles), money launderers, anonymous pranksters, spammers and vigilantes, engineers of walled gardens and those who want to copy and save everything.

Choice is not a panacea. In privacy, we have been burned by the promise of empowering the consumer. Empowering the consumer has led to market failure, not to privacy. Users do not know what to pick, and may not choose wisely. Or perhaps their true preferences may not match what we want for them. Choice may not even be good for innovation. Sometime the best spark is to take away something. The PC motherboard was a big constraint, with just a few slots to add a component, but look what it triggered. A totally plastic environment in which anything can change may not be stable enough for innovation.

For those who think computer science does not design for values, efficiency is our value. This is a deeply embedded value that pervades much of our design.

One of the advantage of doing scenarios is that by walking through the case carefully "from the beginning" one may uncover steps that are important to the success of the story, but which have been left for later development in the research. Questions such as "who creates IDs and key pairs, and what power do they have" may not come up until you step through a scenario.

Returning to the scenarios

1) **Reporting a protest.** This scenario was discussed in some detail as part of the consideration of Question 2 on control.
2) **Facilitating broadband deployment and user choice.** None of the proposed architectures have features that really mitigate this problem. Facilities are costly, and tend to be installed by large players. Architecture cannot change the basis economic reality of cost recovery. The FIA architectures offer two possible benefits. First, the provision of new services may pump new money into the ecosystem. One would have to speculate whether the magnitude would be sufficient to change the overall landscape of facilities competition. Second, flexible multi-homing and multi-path may allow smaller access players to enter the market, and allow the emergence of ad hoc and peer networks. NDN would allow the carriage of data and interests opportunistically by passing devices that serve as "data mules".
3) **Gambling.** We were given a specific formulation of this scenario, as follows:

> *Online gaming is a multi-billion dollar annual business. It needs very reliable connections, because the enemies of the business are lag (any confusion about the order book), downtime, and any suggestion of unfair advantage or unreliable transactions. It also needs secure payment processing that can handle many currencies and types of transactions.*
>
> *I run an online poker operation, incorporated in Antigua, with key officers resident in Gibraltar; my servers are on a Mohawk reservation in Quebec, right by the American market on top of a major fiber optic corridor. (Of course, this assumes present-day geographically bounded client-server systems, but we can posit alternative systems). 90% of my customers are in the United States (where my operation is illegal). I have deep pockets to pay for robust services and paths on the network, and my diplomatic situation prevents the U.S. government from simply arresting my company's officers or seizing my servers. My customers are willing to go to considerable trouble to access my services, but they are by and large "every-day" users.*
>
> *I have two threats: one is my competitors who want to use exploits (like DDoS attacks) to drive me out of business or otherwise capture my customers any way they can. The other is the government, which just wants me out of business, by seizing my domain names or other human-readable addresses (without a centralized DNS, though?), by making it impossible for me to settle financial transactions, by rendering my service unreliable (perhaps by non-neutral data carriage), or by some means peculiar to a future architecture.*
>
> *In your system, what's my ideal strategy? And what's the government's ideal strategy to take me down? How can they regulate or exert control, and how can I counteract them?*

Nebula, with its emphasis on a highly reliable and robust core, would seem to work to the advantage of the gambling industry and against the interests of the governments. The emphasis on controlled latency would seem to favor architectures that allow for explicit service setup. NDN, with its lack of host names, would seem to deflect some of the DDoS attacks that have been launched against some of these gambling sites.

4) **Just and fair treatment in times of disaster.** There seem to be two aspects to this scenario. The problem of basic connectivity benefits from architectures such as MF that stress dynamic multi-homing. Diversity in access is an objective that is consistent with several of these schemes. The second aspect of the scenario, that emphasizes the "just and fair" aspect of the challenge, seemed to be beyond the scope of these architectures, as it rests more at the application level.