BBC

R&D White Paper

WHP 011

November 2001

Control and asset management in future networked studio architectures

P.N. Tudor

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper 011

Control and asset management in future networked studio architectures

P. N. Tudor

Abstract

This white paper describes the work and conclusions of the Pro-MPEG Forum's Networking & Control group, chaired by the author. It was presented at an IBC 2001 mini-conference entitled "Networking of TV in Production" sponsored by the Pro-MPEG Forum and the AAF Association.

The work is presented here as slides with notes.

Key words: television, production, network architecture, Pro-MPEG

White Papers are distributed freely on request.

Authorisation of the Head of Research is required for publication.

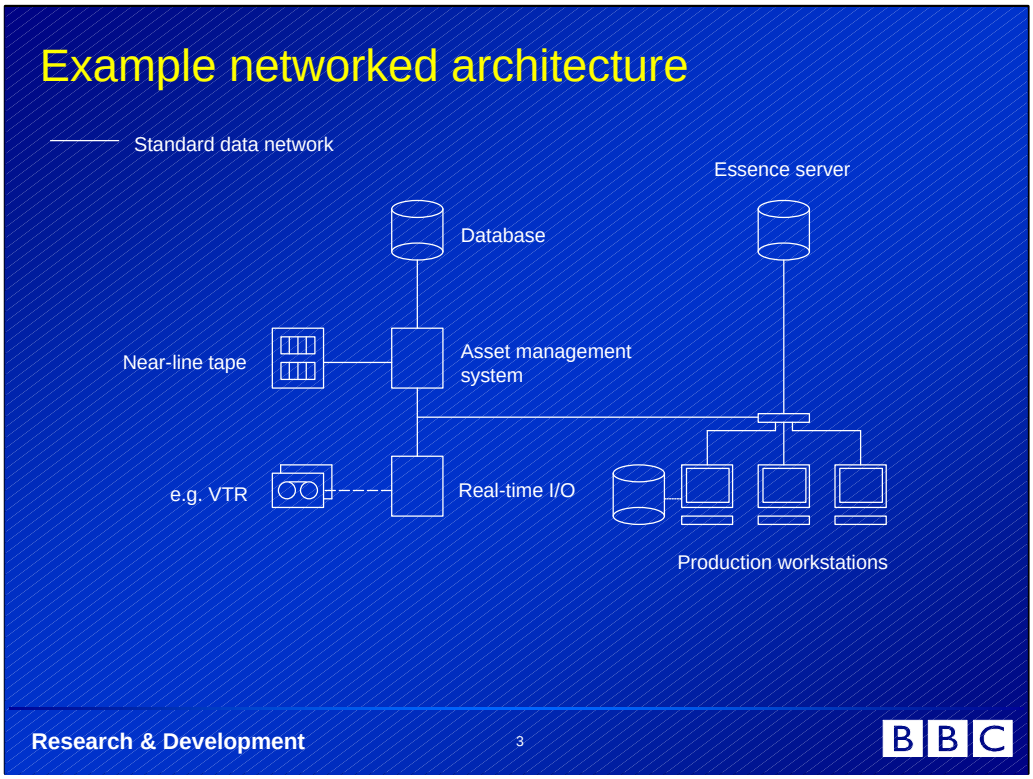
Control and asset management in future networked studio architectures

Networking of TV in Production, IBC 2001

Phil Tudor, BBC R&D, 14 September 2001

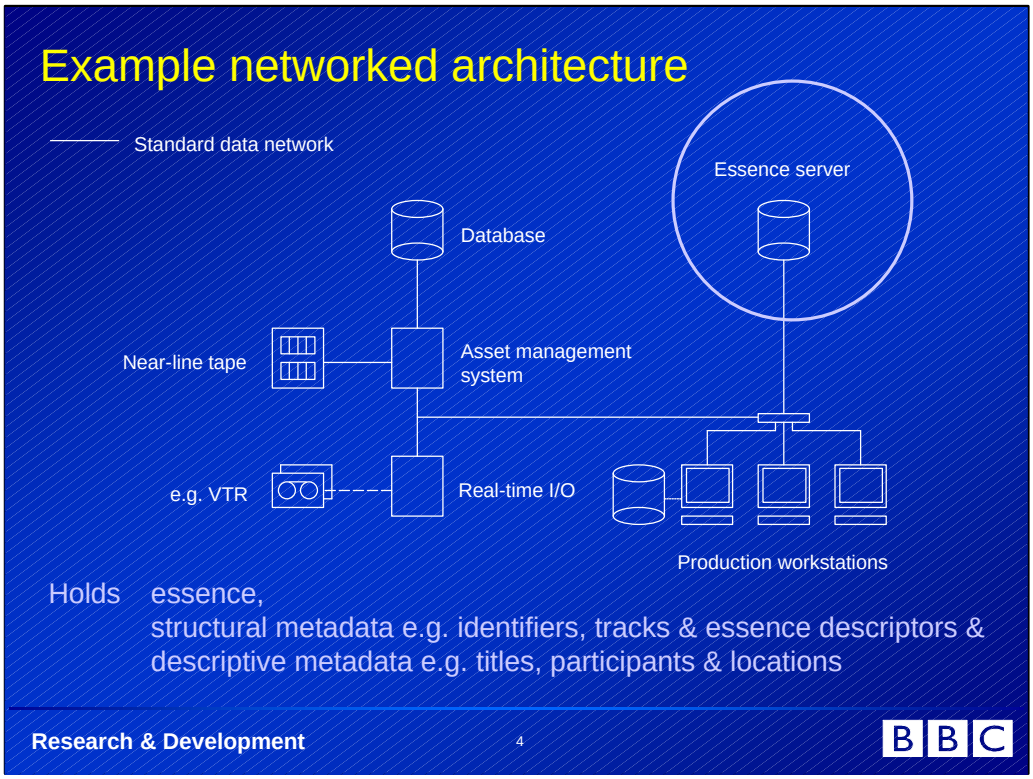
Definitions

- Essence
  - directly perceivable data e.g. video, audio & data services
- Metadata
  - data about essence
  - data that is not essence
- Corollary
  - “One man’s essence is another man’s metadata”



Here's a schematic of an example networked architecture.

Let's point out a few features...



The essence server stores:

the essence;

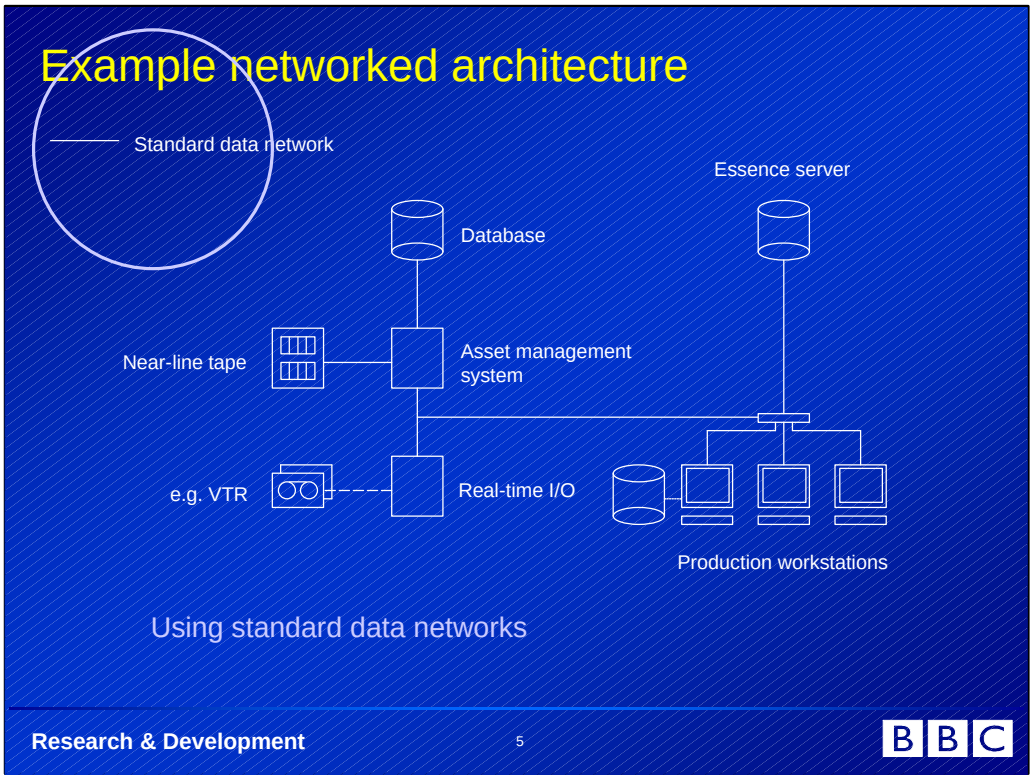
and if we're using a format like MXF* or AAF*, there may also be structural metadata in there. Some good examples of that are unique identifiers, the number of tracks and descriptors for the essence;

and descriptive metadata. This is editorial information such as titles, participants and locations.

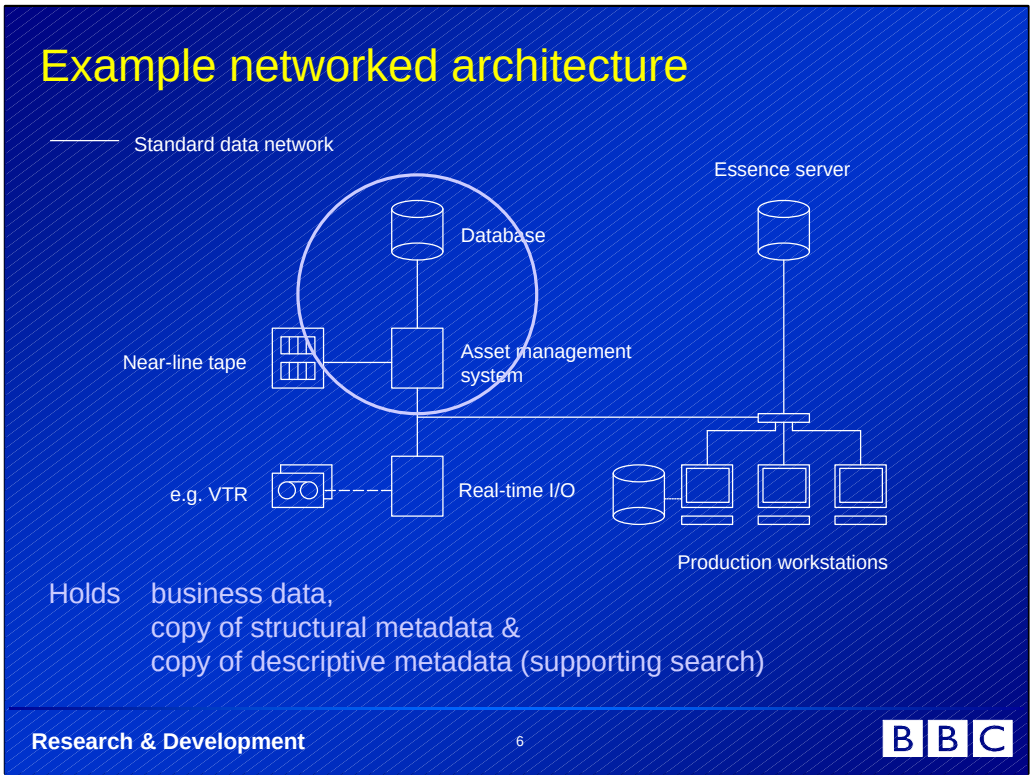
The metadata would be there by virtue of being bundled up in an MXF or AAF file.

* MXF = Material eXchange Format (from Pro-MPEG Forum, AAF Association and EBU)

* AAF = Advanced Authoring Format (from AAF Association)

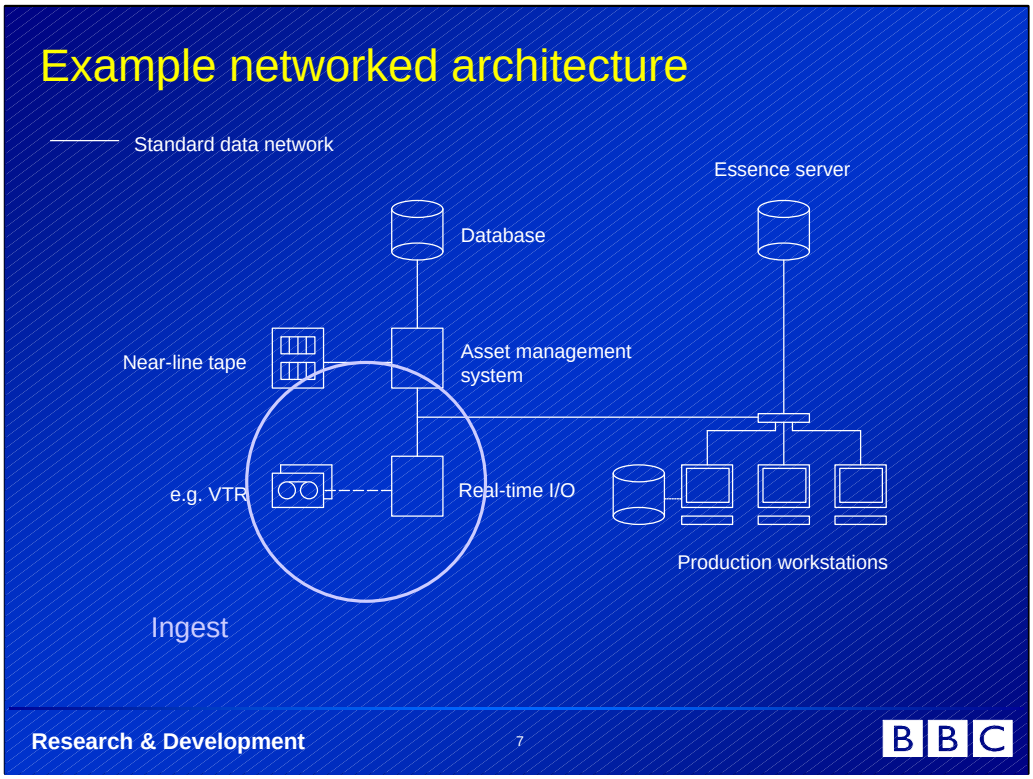


We're looking slightly ahead to an architecture and a system built using standard data networks. These might be fast ethernet, gigabit ethernet or fibre channel. But possibly not things with BNCs on the end.

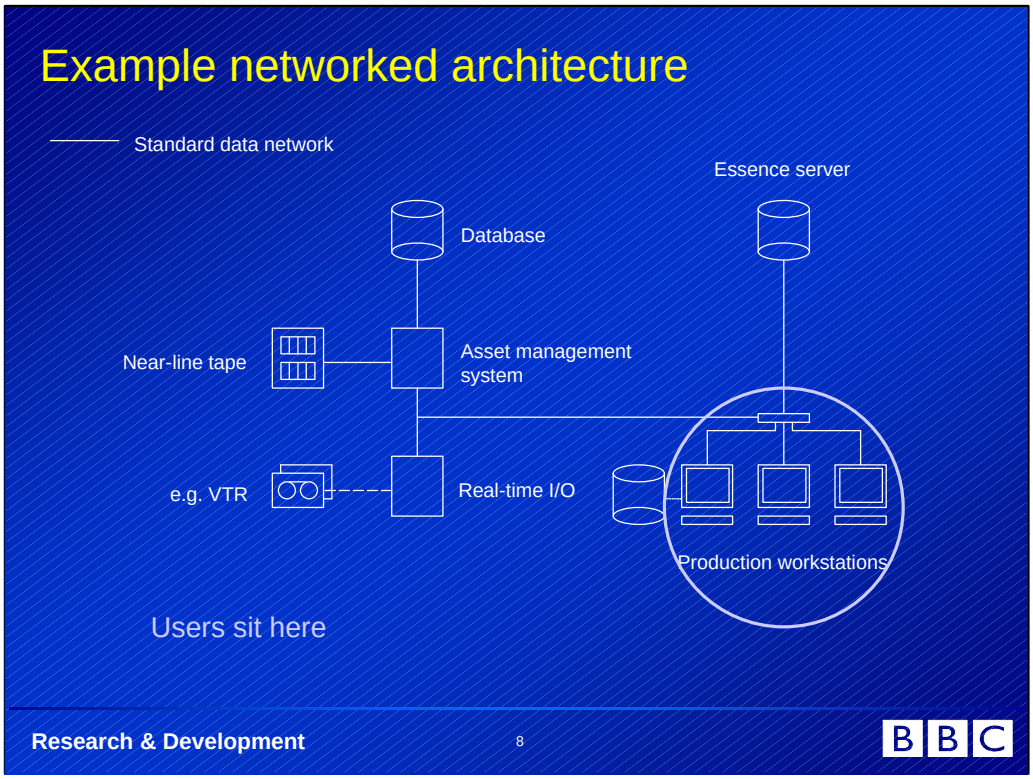


The asset management system is the heart of the control system. This holds additional business metadata. For example this system knows about the people working on the system.

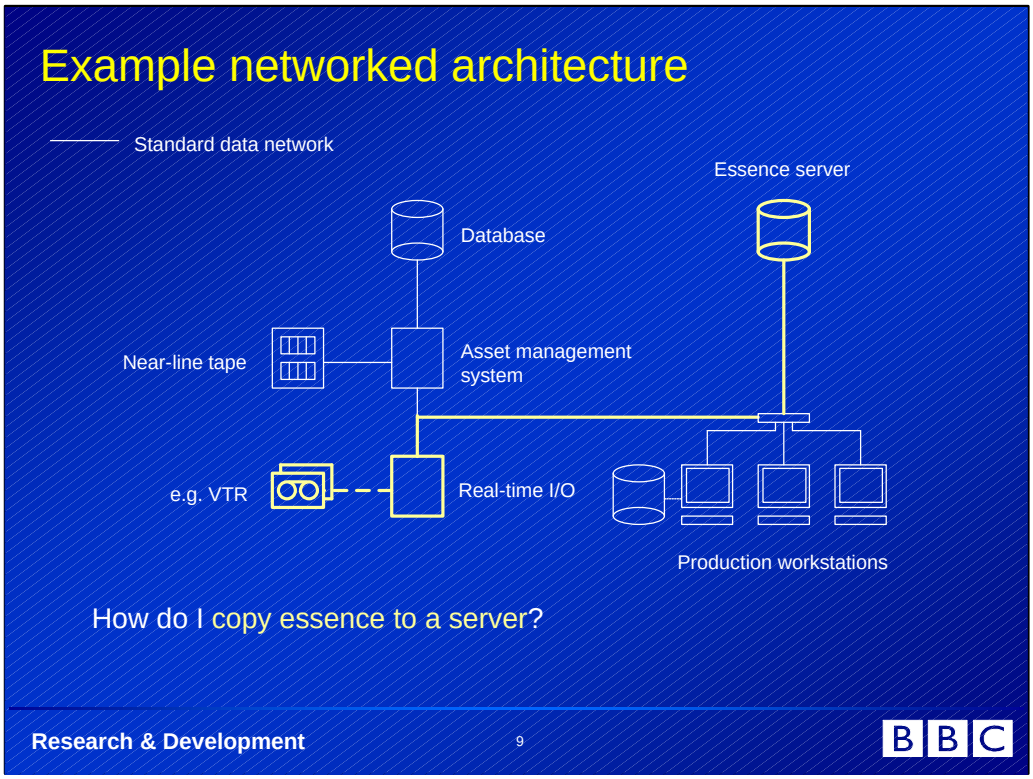
It also has copies of the structural and descriptive metadata from the essence server. This is copied to the asset management system to support searching and querying on that metadata.



Ingest is a way into the system—an interface between real-time streaming and networked file transfer.

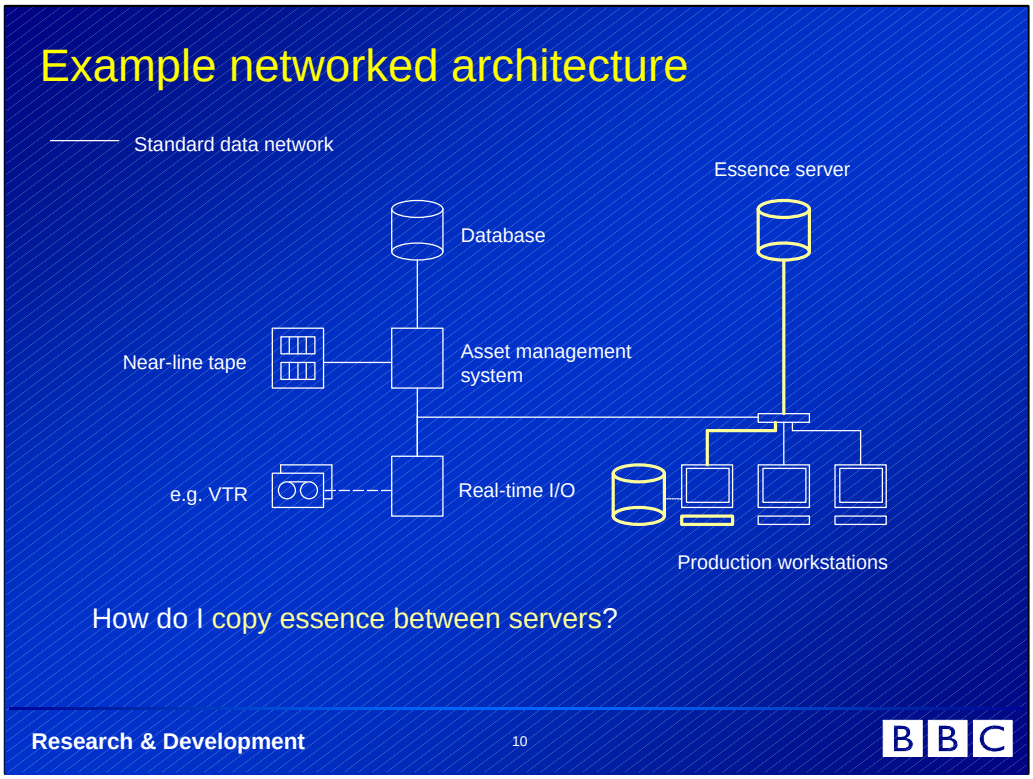


Finally this is where the users sit. Production workstations are used to access all the networked facilities.

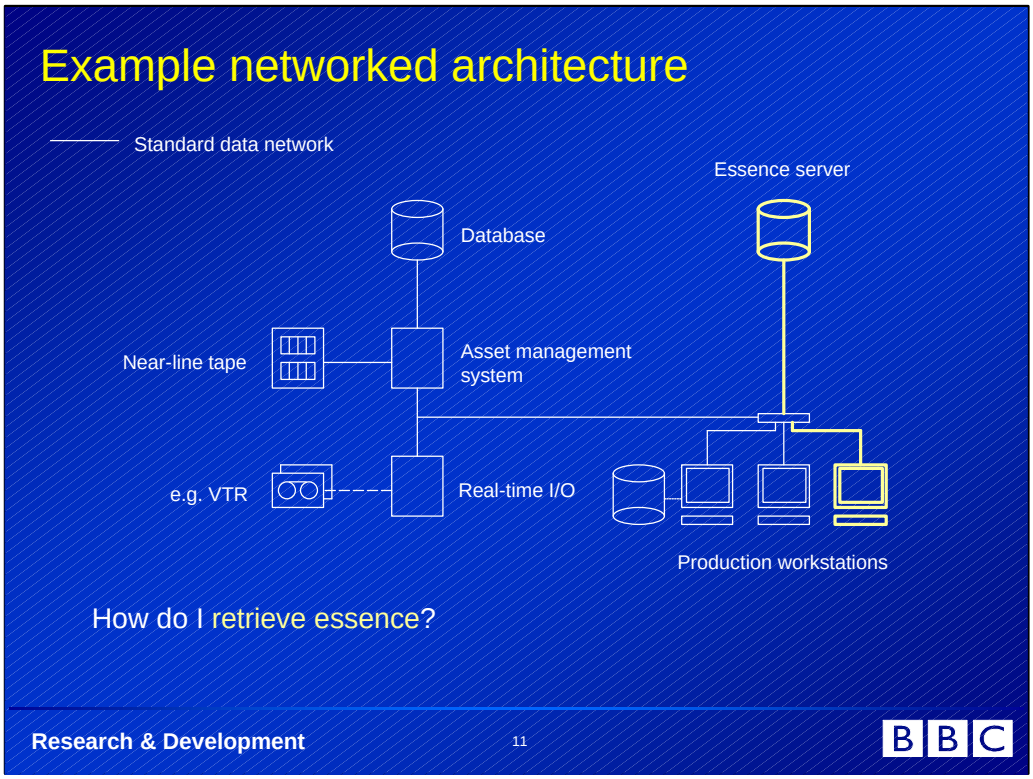


So what about the networking & control issues?

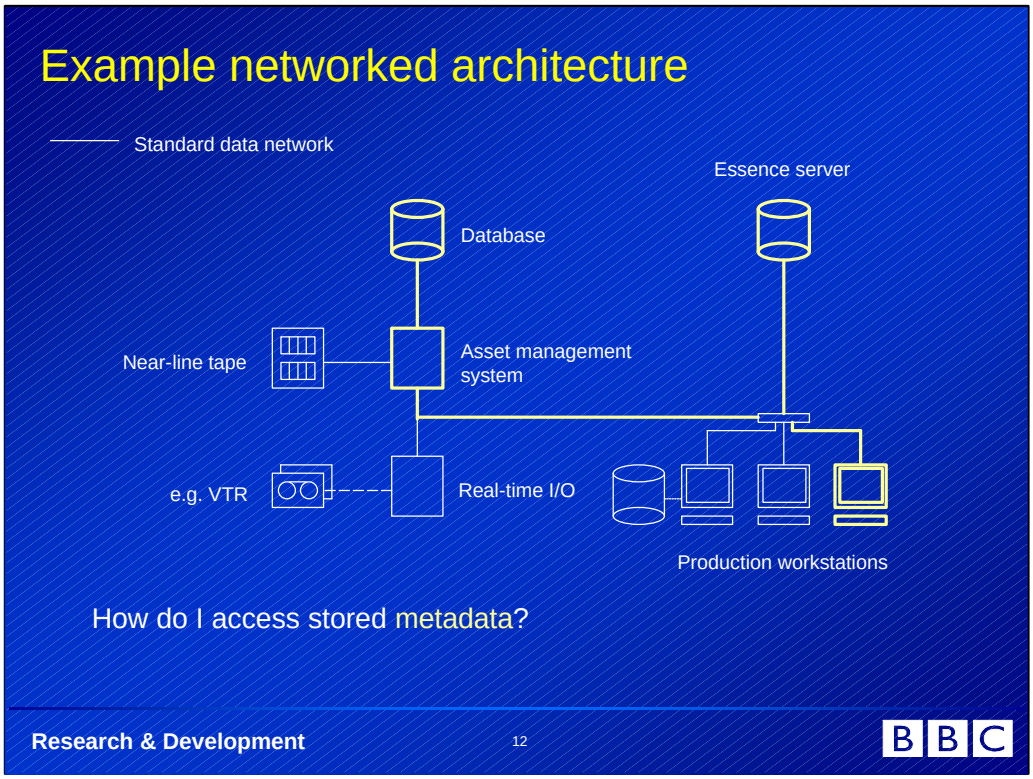
Even given MXF and AAF (which define data formats), what protocols do we use to transfer essence from a real-time device, across the network and on to a server, assuming we have a mixture of components from different vendors?



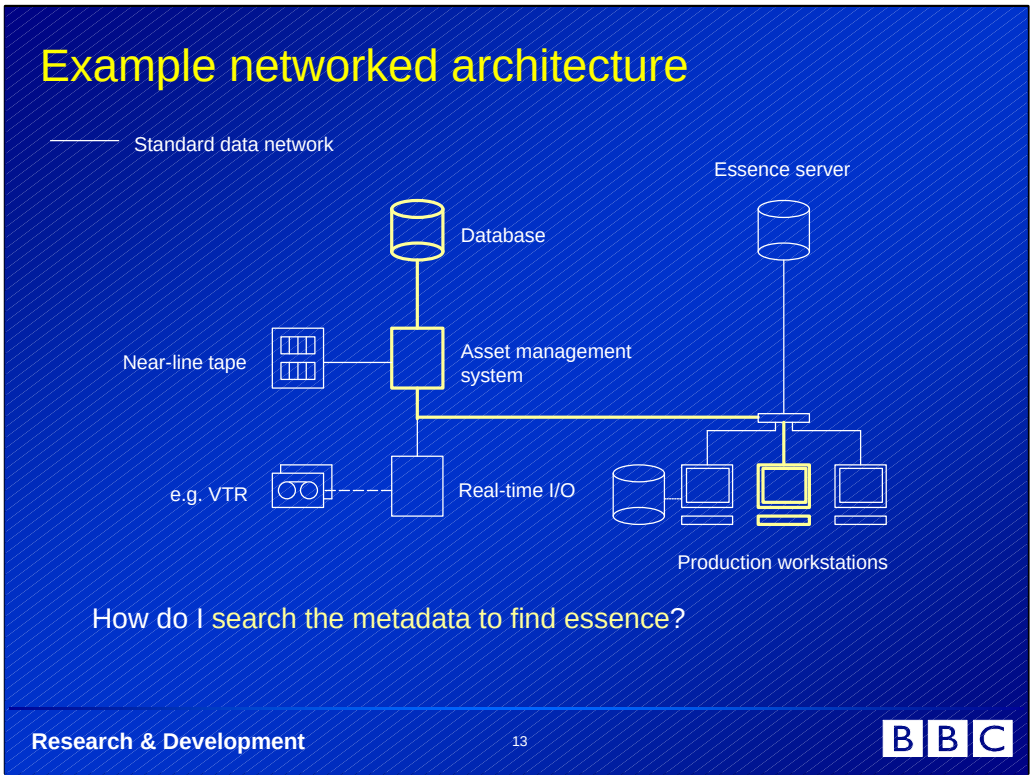
How do we copy essence between two servers?



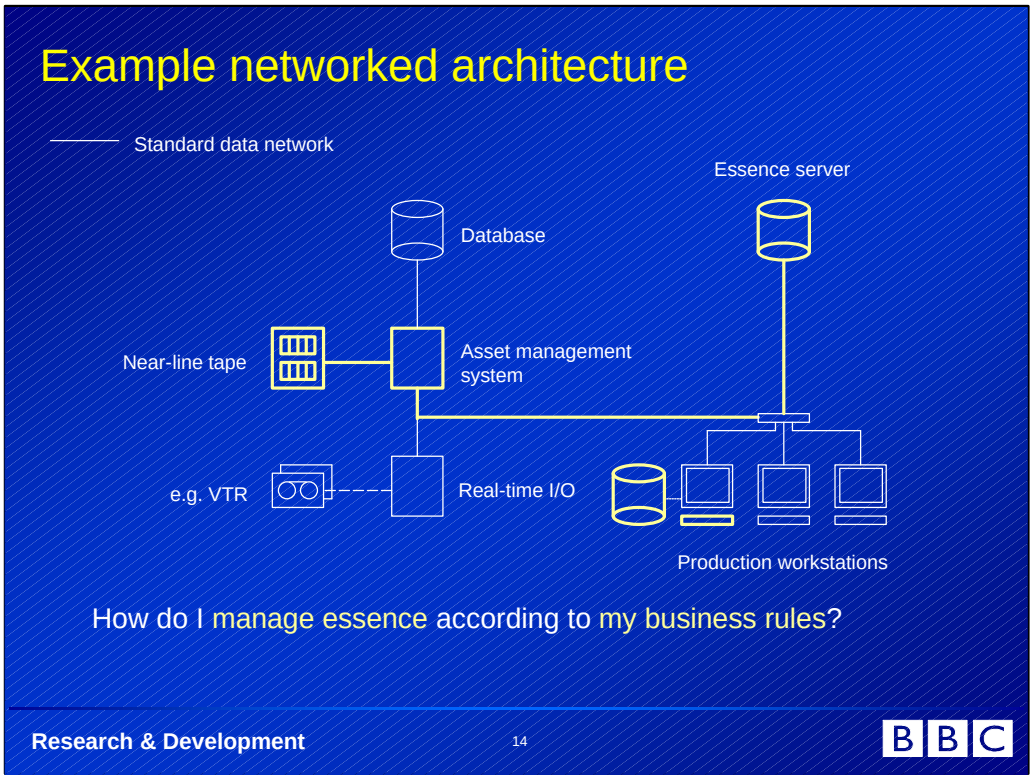
How do we retrieve essence, for example bringing it to a workstation and displaying it?



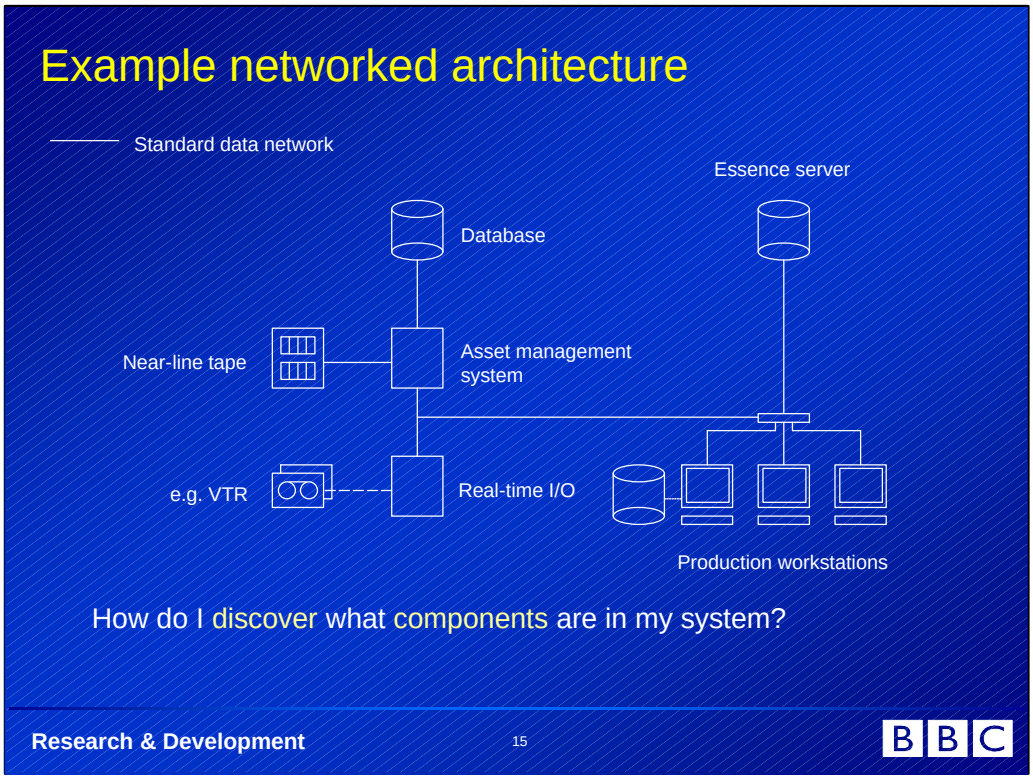
How do we access the metadata that's stored remotely across the system?



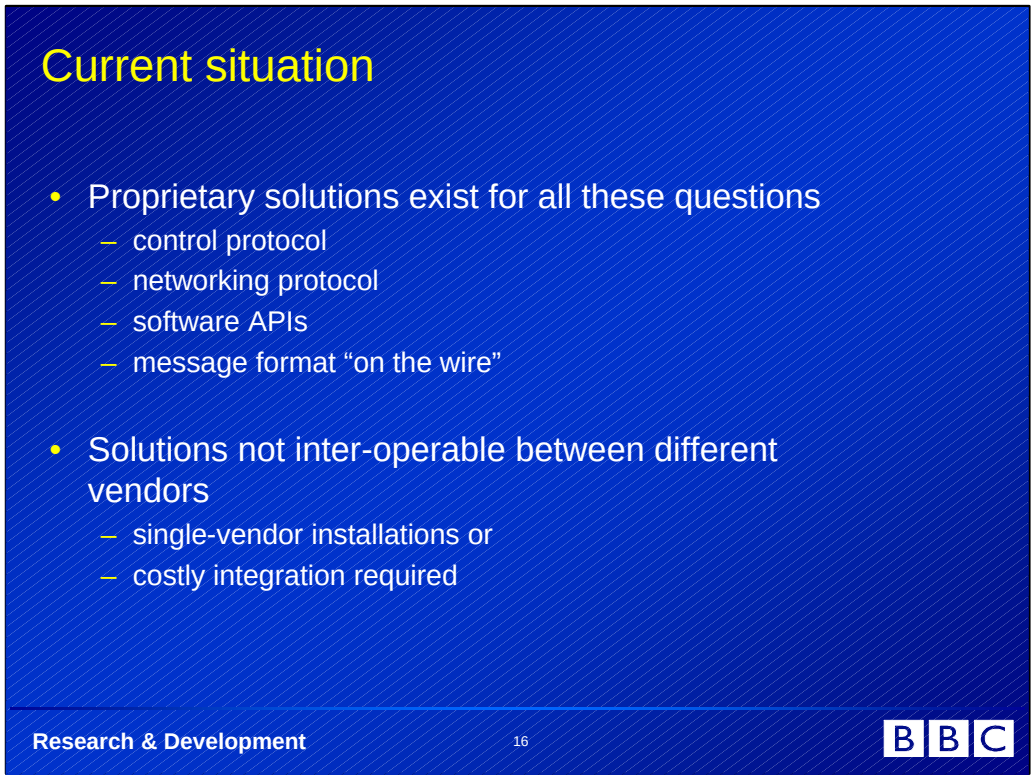
How do we ask for searches to be done?



How do we implement business rules such as archiving policy and controlling access to our essence?

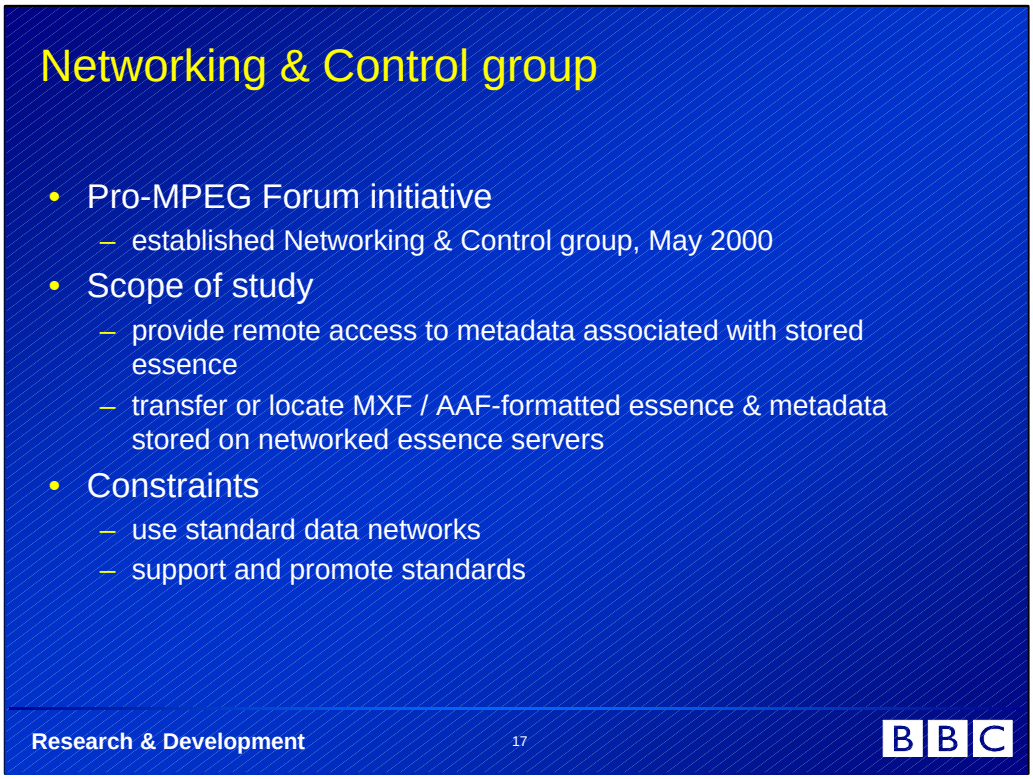


How do we even know what components are in our system?



The current situation is that there are proprietary solutions to all these questions. Those solutions consist of control protocols, networking protocols, software and formats used on the wire. The problem is that those solutions are not inter-operable between vendors.

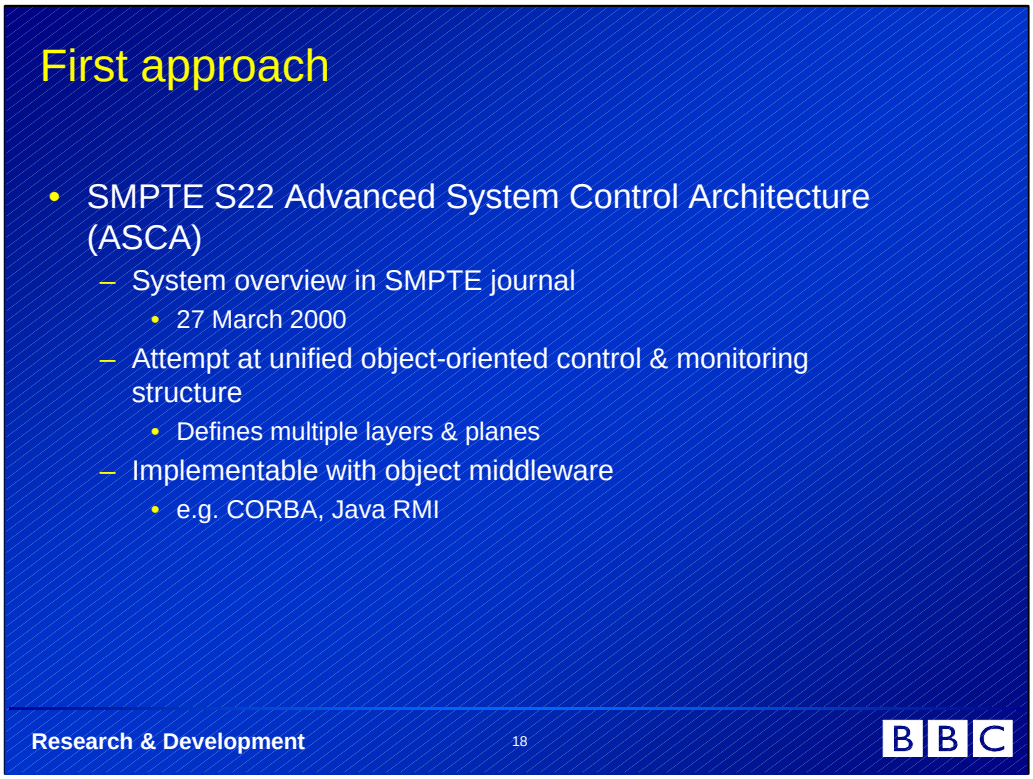
The consequence to a user is that we are either face single-vendor installations, or spend a lot of money on system integration.



The Pro-MPEG Forum created a Networking & Control group in May 2000 to study this problem.

The scope of study was kept simple: look at ways of providing remote access to metadata associated with stored content on the network, for example in order to choose a particular remote file. Then having chosen a remote file, transfer it or at least locate it, perhaps using a URL.

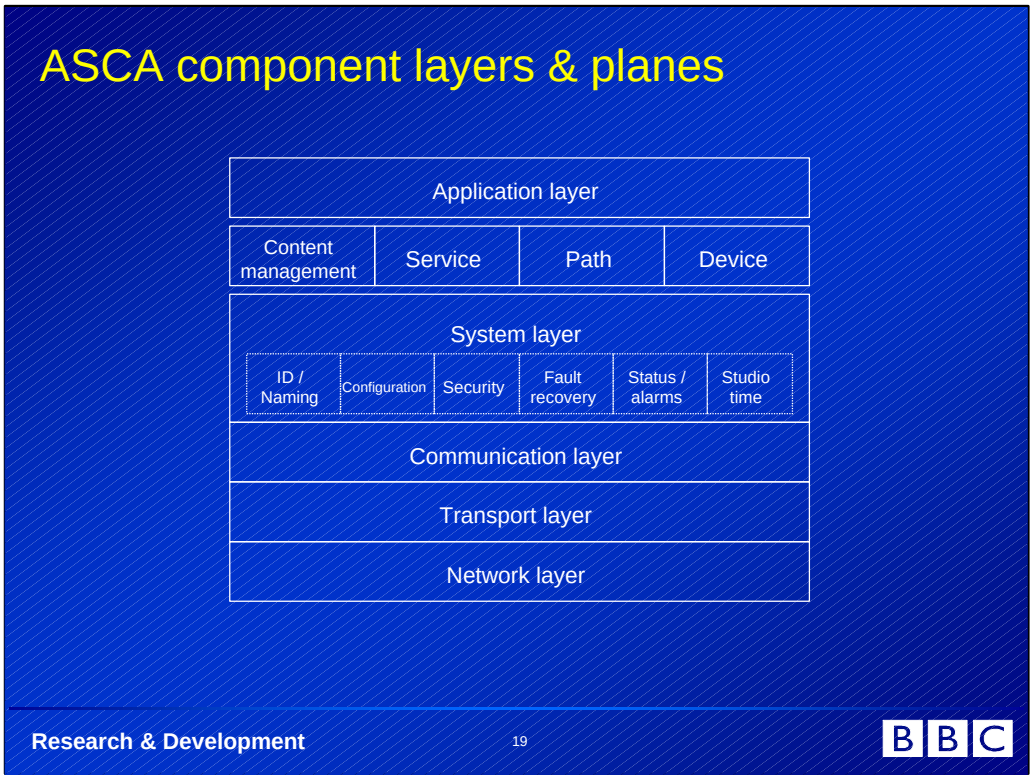
The constraints were to use standard data networks and to support and promote standards where they existed or were being developed.



This was the first approach.

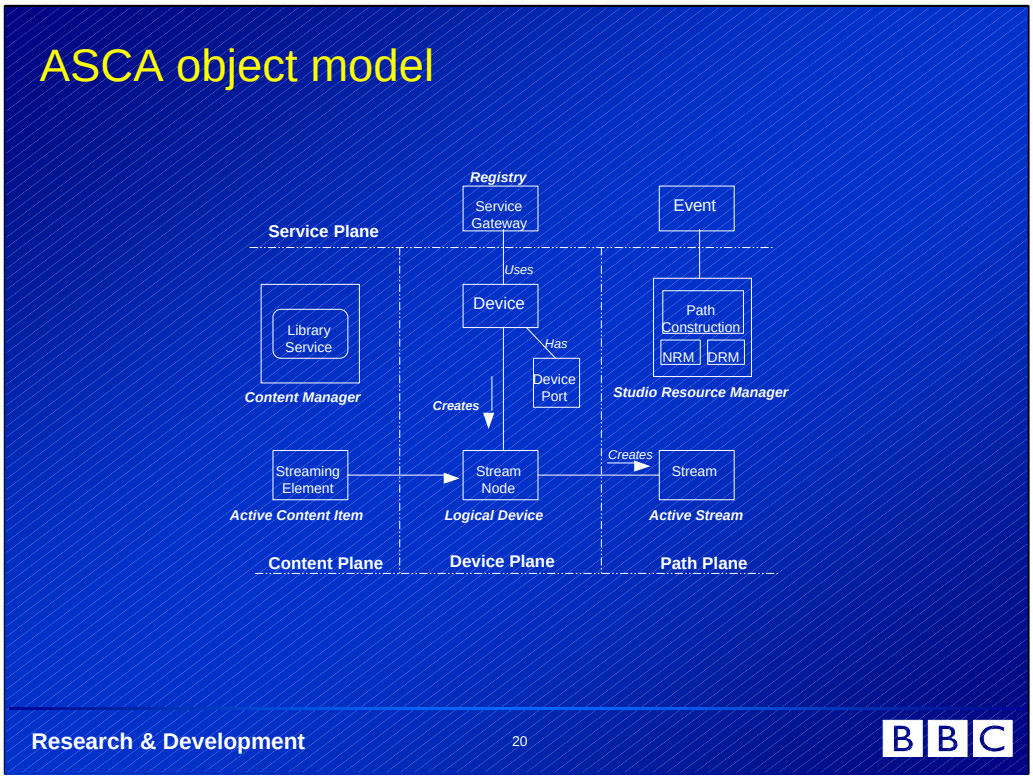
The SMPTE, in their “task force final report”, had identified similar problems. They created a group called the Advanced System Control Architecture (ASCA) group.

The ASCA group undertook a comprehensive study of this area with a fair amount of industrial input. They produced a structure in which many different aspects of control and monitoring could be unified. The design was object-oriented and could be physically implemented using object middleware such as CORBA or Java RMI.



This is a diagram of the ASCA structure, simplified from its original "cube" to a set of layers. It has the usual characteristics of layered systems—a layer makes use of the layer beneath it and provides services to the layer above it. Many control and networking aspects can be expressed within this over-arching structure.

The second layer from the top was actually a set a planes in the ASCA original. Starting at the right-hand end: there is the plane for device control; then the path plane for setting-up connections; then the service plane for aggregating connections into services; and finally the content management plane for accessing metadata.



This is the object model of the ASCA work.

On the left is the content manager—this is the library service where you go to do searches.

At the top is the service gateway, for component discovery. If we plug a new production workstation into the system, it accesses the service gateway in order to access every other component in the system.

The resource manager manages the finite resources of the system. We don't just take resources, we ask for them through the resource manager.

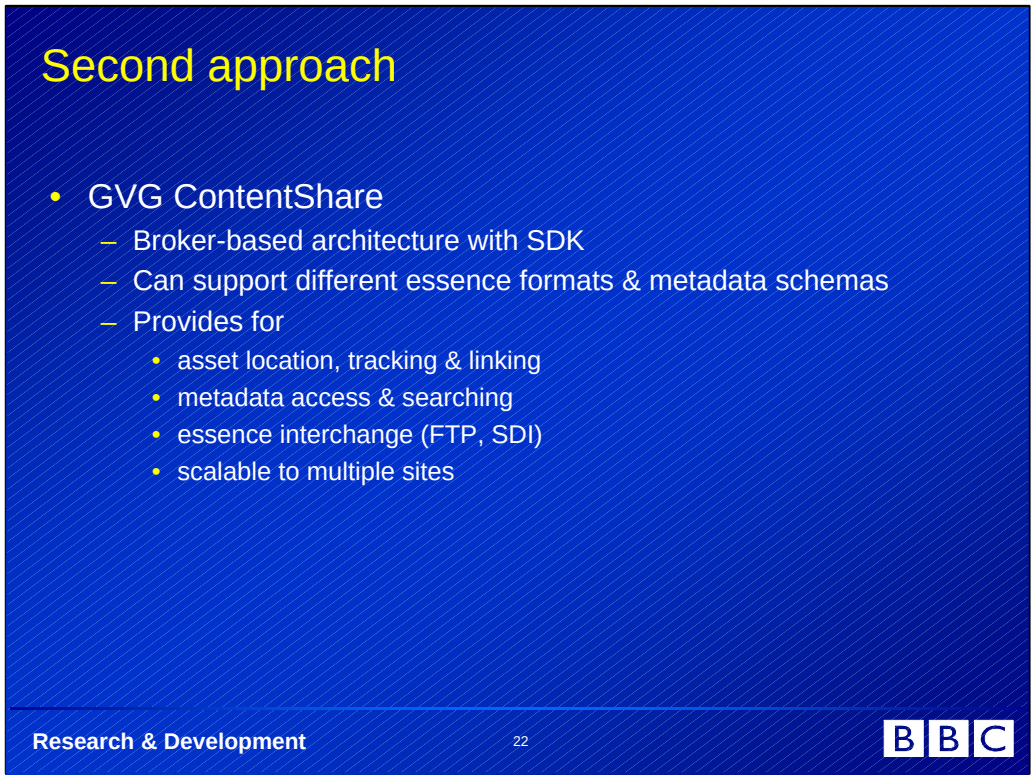
At the bottom, we have streaming elements (which is the stored essence), the stream nodes are ports on devices, and a stream is an object representing a transfer in progress.

Results of ASCA approach

- Defined UML object interaction diagrams for
  - library service registration with Studio Resource Manager
  - essence server registration with Studio Resource Manager
  - production workstation searching library service
    - examining metadata
  - file transfer between essence servers
- Implementable by re-using existing IBM & Sarnoff code
- No take-up by Pro-MPEG industrial members
  - SMPTE ASCA group ceased

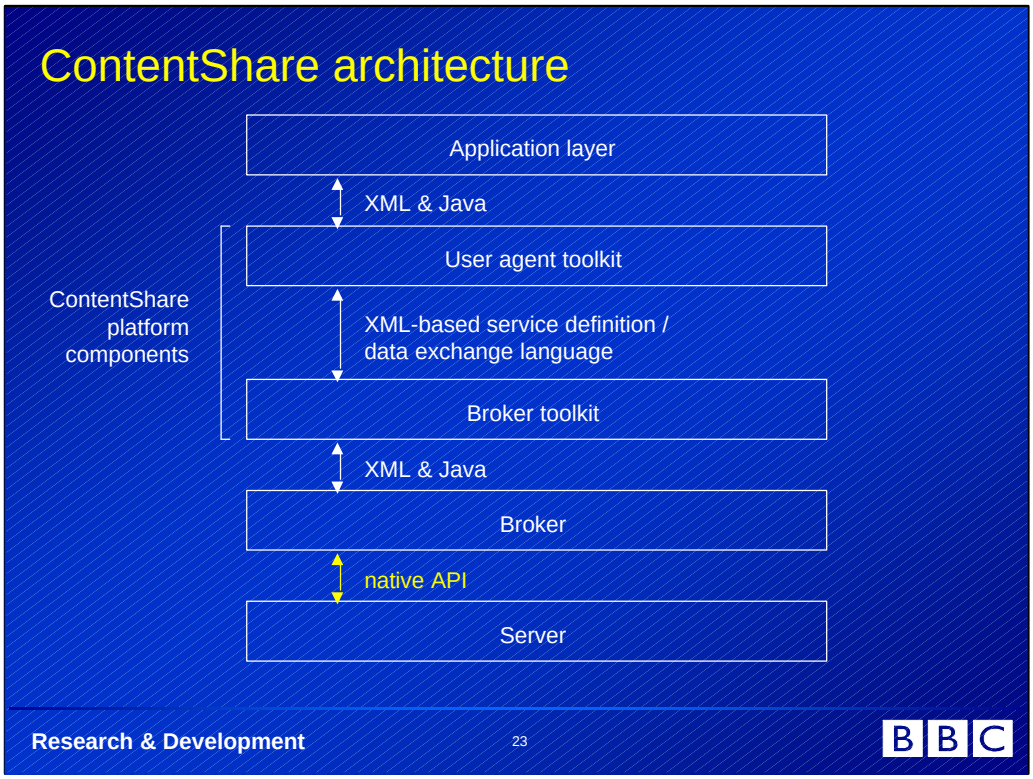
The Pro-MPEG Networking & Control group took a thin slice of this large work to cover the aspects we were interested in. Using UML, our slice included discovery of components, interfaces for searching the library service, and initiating file transfers. The implementation plan was to use code from IBM and Sarnoff who had prototype systems. That's the good news.

The bad news is that there was little take-up within Pro-MPEG's industrial members. And a similar problem occurred in SMPTE so their initiative ceased. Although this was a useful experience, it was ultimately a dead end.



This was the second approach.

Grass Valley Group have been developing a system called ContentShare. This has similar aims and goals, in my opinion, to the ASCA work, albeit from a specific manufacturer. It is a set of software brokers and a large software development kit. It's independent of essence format and particular metadata schemas. It provides these sort of facilities: asset location, metadata access and searching and essence interchange as FTP or baseband SDI.

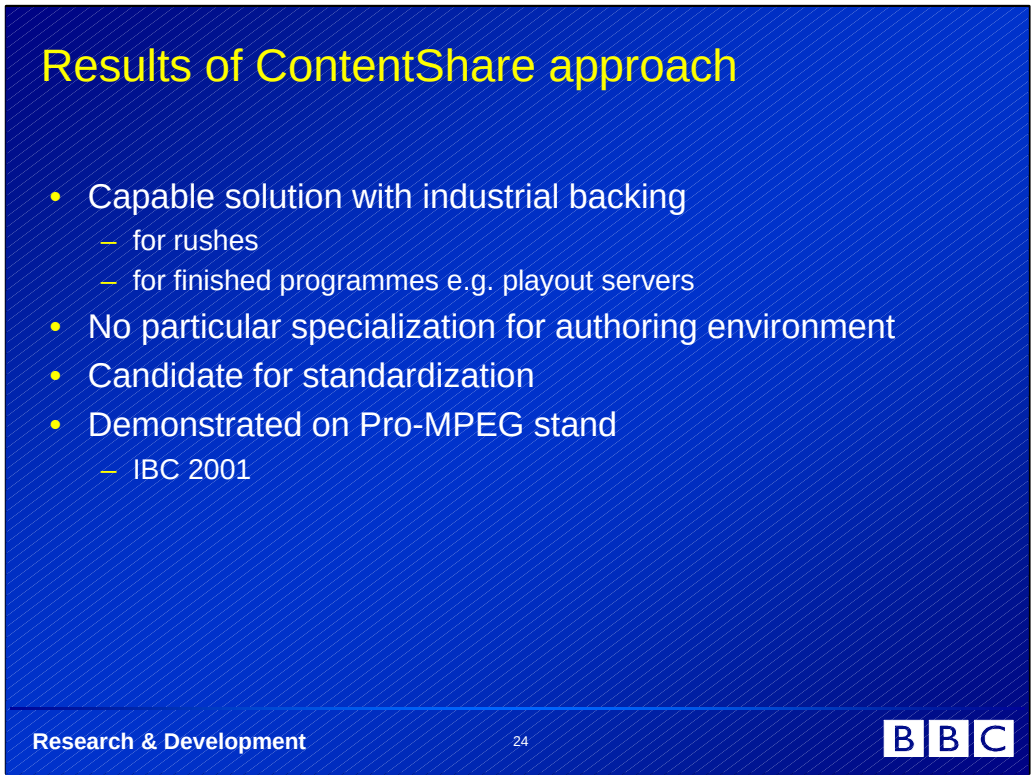


This is a diagram of the layers within ContentShare.

At the top is the application and at the bottom is the essence server somewhere in the world—in the middle is the ContentShare platform to connect the application to the server. The ContentShare platform is provided when you sign up to ContentShare.

The magical properties of XML and Java are used essentially as a control protocol—a flexible way of transferring metadata and control information between components and also for servers to describe what sort of services they offer and what types of metadata they're going to be producing.

To integrate a server, you have to write a broker to map between the server's native API and the ContentShare platform.



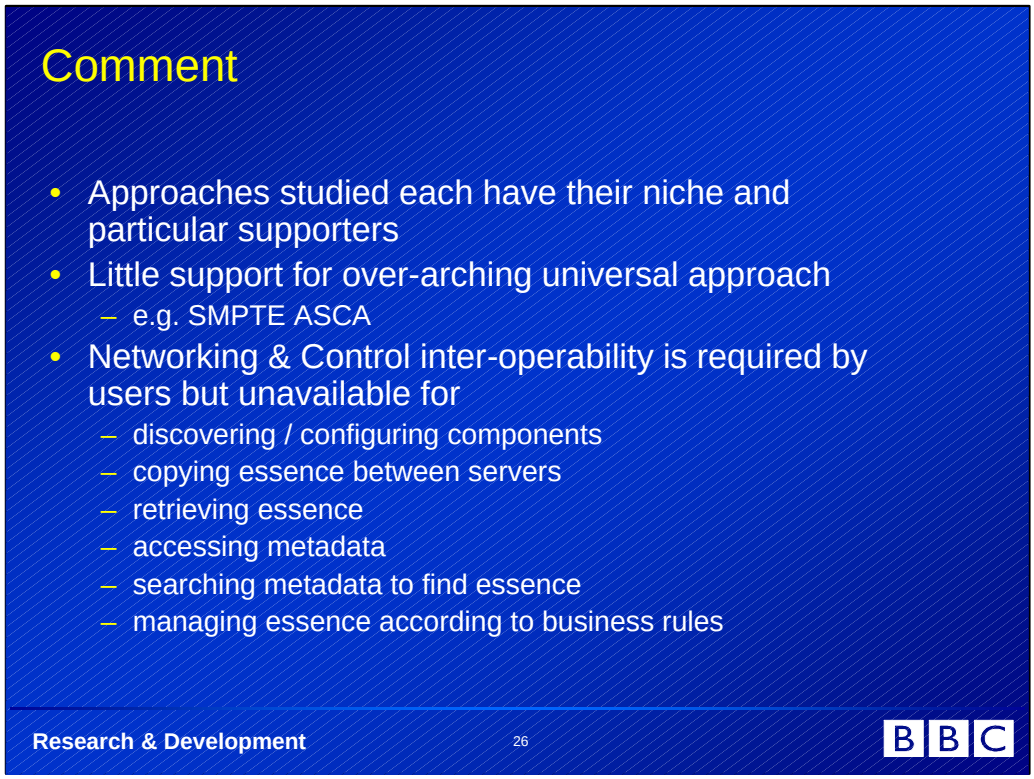
ContentShare is a capable solution with industrial backing. It is strong on rushes and finished programmes but doesn't have particular specializations for authoring environments. Its promoter is considering it for standardization and is shown on the IBC 2001 Pro-MPEG stand.

Other approaches:

The MOS Protocol from AP. This enables servers to list assets to a central control system, typically ENPS, a specialized system for News. The problem there is that, having collected all this information, it is only available to ENPS clients. For example, there aren't open interfaces into ENPS for searching for content.

Avid have MediaManager which is an asset management system with the same object model as MXF and AAF. Their Transfer Manager moves files around a network. These are two components of their Unity system.

And of course there's the ubiquitous web-site and many asset management companies' products are basically web-sites. The problem there is that although they're inter-operable in the sense that you can point any web-browser at the web-site and manipulate the system, you haven't got an open interface you can go to and get specific services.



The different approaches all have niches and supporters.

There was little support for the over-arching universal approach that SMPTE tried. This was a laudable piece of work and was strongly supported by users, including the BBC, but there is little support from the industry at the moment.

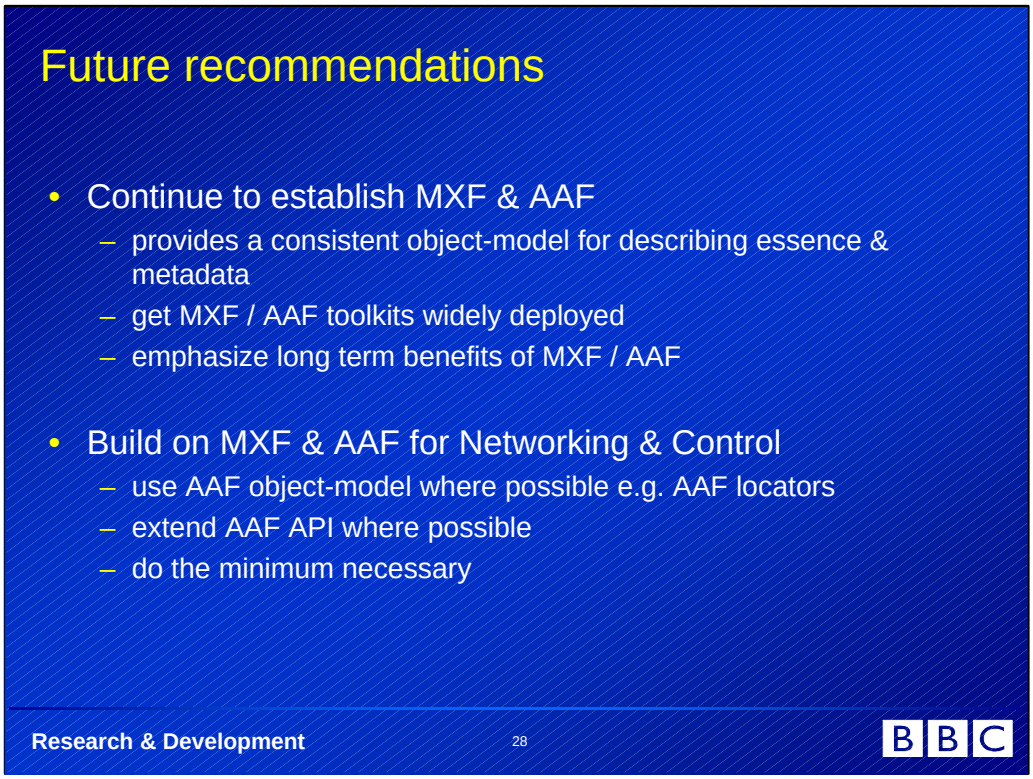
The sort of networking & control inter-operability required by users to make the efficiency gains they hope for is unavailable at the moment.

What are the lessons to be learnt from this?

Entirely new big solutions are not attractive to industry. Anything that requires significant time to understand is immediately expensive. There's a fear of hidden snags lurking beneath complexity—"is there something in there that's going to disadvantage me when I actually get down to it."

A lot of these systems are complicated which leads to associated software toolkits. Unfortunately, vendor A's software toolkit is usually unattractive to vendor B.

Users are demanding more inter-operability on commodity mass-market hardware for less money at the time when the industry's suffering an economic downturn.



So what should we do?

We should tackle one problem at a time. I feel we've been asking industry to look too far ahead in terms of defining standards.

My recommendation, and the output of this Pro-MPEG working group, is that we focus on establishing MXF and AAF first. This gives us a consistent model for describing essence and metadata. As we don't have that today, many systems invent this again and again. We need to get MXF and AAF firmly in the market first.

There are long term benefits to adopting MXF & AAF. It solves the file format problem today but also provides a foundation for the features I've been talking about in this presentation tomorrow. For example, the AAF object model already has features to do with distributed content (AAF Locator class). Adding new features to models already understood by software developers is an attractive proposition.

The final thing is do the minimum necessary—not a very good note to end on! You're on a hiding to nothing if you try and put a gold-plated solution in. You have to do one thing at a time, get it accepted and keep it as simple as possible.

Thank you

phil.tudor@rd.bbc.co.uk