

# Does anyone have a pattern for...?

Or, designing loads of things at the same time and how to keep them consistent...



Vicky Holland  
Senior User Experience Architect

Hi! 🖐️

Hihi, thanks for coming

I'm going to talk a bit today about how as a member of the team responsible for designing internal tools at the BBC we try to work in the most consistent and efficient way possible and also how in my role as a UX architect I design information architecture that supports the design systems we use.



SYSTEMS + SERVICE DESIGN

So today I'm going to talk a little bit about:

The team I'm in, what we work on and what unique challenges we face

How we develop and maintain our design system

And how good information architecture

should support that design system

As the title suggests, one of the most common questions in our team slack is does anyone have a pattern for such and such - we're a small team so making sure we're not duplicating work is really important, hence the importance of having a design system

So I work in the Systems and Service Design team at the BBC - we are the team responsible for designing all the systems that staff use. The team consists of about 20 people - UX designers, UX architects and Researchers and Creative Directors and we all have different specialisms that we bring to the work we do as a team

The service design part of our name comes from the fact that we have to work with multiple teams, products and stakeholders to deliver joined up services for our users.

## **ABOUT 700 TOOLS**

How many tools do we have?

Sounds like a lot but the BBC has over 20 thousand employees who do a huge variety of jobs

These tools generally fall into two categories - what we call enterprise tools (hr, room booking, procurement, holiday) and production tools, which

are basically anything needed to create content and get that content to out audiences (so content management systems, libraries, broadcast systems)

# 120 PRODUCTION TOOLS

Today I'm going to talk mainly about how we design for production tools, of which, fortunately there are only about 120

Most of our production tools are very specific pieces of software that have been built by in house teams sometimes in quite a siloed way, this is as opposed to the enterprise systems



which are often bought in and have limited scope for customisation.

Production tools are not just web based things, sometimes they're pieces of software that support specific broadcast hardware, but generally they're online tools built for specific products, teams or use cases.

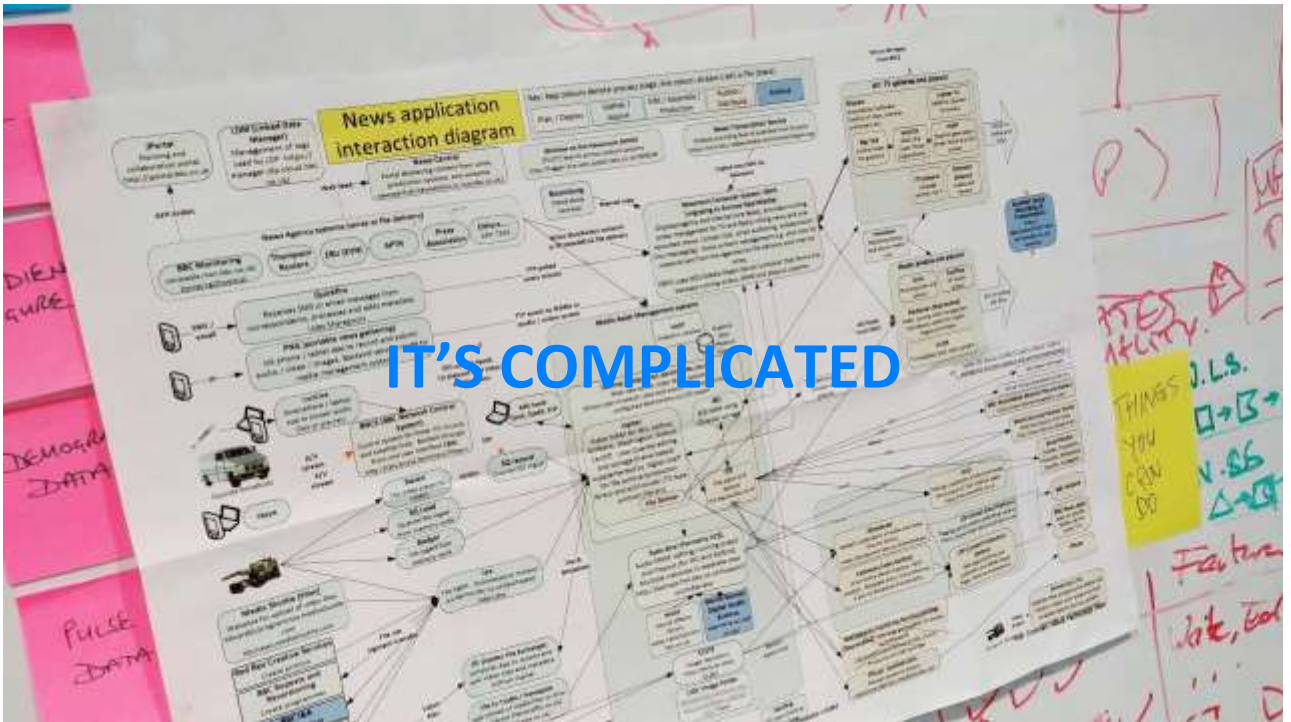
Some examples include, article editors, image editing, automatic subtitling, video clipping, editing tools, and tools for broadcast for eg. screens showing incoming feeds from broadcast trucks or tools that help people schedule content.



This mix between broadcast technology and online technology makes it both a really interesting and really challenging area to work in.

We recently went to visit the sports studios in Salford and I was just totally overwhelmed by the amount of infrastructure still involved in broadcasting.

This is me in a server room, which is apparently one of about 40.



Because we have such a large number of tools that we can potentially work on it's really important for us to have some kind of design system. There are literally not enough of us to design every single part of every tool. This also means that we have to be much more strategic in our approach to designing.

Lots of tools basically means, lots of stakeholders, lots of different roadmaps, and lots of complexity.

It also a makes it difficult to work towards a shared vision or strategy, or to share or reuse work across multiple projects. This sometimes results in a user experience that is not as joined up as it could be!

**PLANNING TOOL**  
**APP FOR GATHERING VIDEO**  
**VIDEO TOOL**  
**EDITING TOOL**  
**VIDEO TOOL**  
**VIDEO PUBLISHING TOOL**  
**ARTICLE PUBLISHING TOOL**

A common example of this is the workflow for creating videos – I'll just show a few of these tools

So journalists start by planning what they need for the day's stories in a planning tool - here they work out what film they need, interviews, how many people need to go out with cameras etc

They might then go out and gather video with their phone and send it back to a BBC site via another app.

Once they get back to the office they can find their video and add metadata to it - this is in another tool, at this point they might they also editing the video in specialist tools to create the package they need.

Then they upload it back to the video tool.

And then, if video is going to be added to a page it has to be uploaded elsewhere so it can have a UID.

The journalist might then create their news

article, add the video UID and publish it (again in another tool) And finally it's available to people!

In addition to the number of tools there isn't necessarily integration at all points, so the user moves to a different tool but the information they have about their content sometimes gets lost or has to be re-entered.

So this workflow involves a lot of different tools that users have learn to and isn't as joined up as is could be

These are some of the problems that we wanted to address by using a design system.





So you probably know that the BBC has a design system called GEL

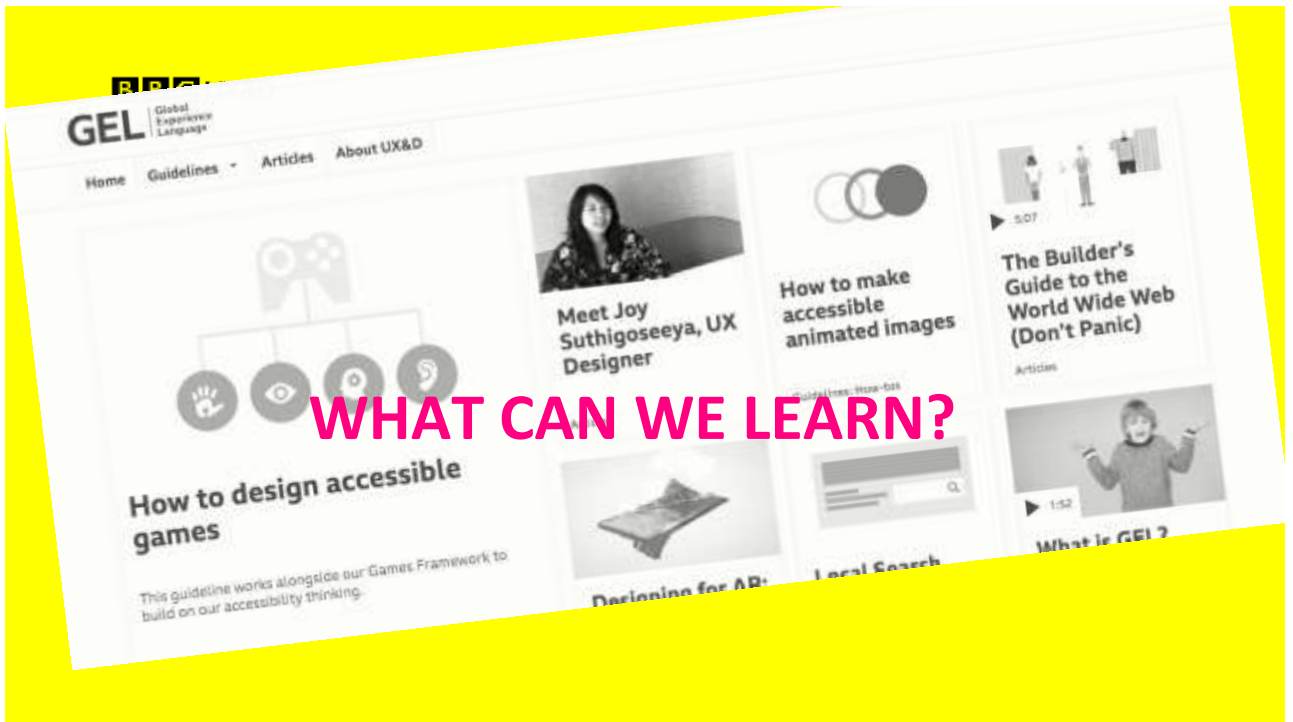
Currently GEL mainly caters for audience facing design patterns, these are used by the UX teams in any of the audience facing products (like news and sport, iplayer, bitesize)

The current GEL patterns tend to focus

on users finding, browsing and reading content whereas the internal tools tend to be more transactional, and focus on forms, lists, quick actions, basically anything that helps users complete tasks as seamlessly as possible.

So the external patterns differ quite a lot from the patterns we need as team where our users are also staff just like us.

That said, there are some really useful things about having a preexisting design system. It basically means that we're not starting for scratch when we start thinking about designing for internal systems, were just adding to what we have by designing for a different set of users.



Some of the benefits are:

General awareness of GEL within the organisation -

We don't have to sell the idea of component based design or the benefits of having a design system to anyone, most of the teams we work with are already well aware of the benefits.

Established language for patterns –  
Because the GEL library has been around for a while now we already have a common vocabulary, this means we don't talk at cross purposes or start designing things that other people are already doing

Preexisting iconography, patterns, grid and typography -

We only need to add to this where our use cases differ from those of the audience facing products

Things we've learnt from GEL

We can use what the gel team have learnt in setting up the design system to inform how we go about setting up our internal design system

Given all those benefits we decided build on the existing work and develop it out into a design system specifically for internal tools.

As a team it was important for us all to be involved because we all have different skills and specialisms and ideas that we can bring to the table.

We facilitated this through collaboration through regular slack showcases and weekly team meetings however it often felt like the work was very piecemeal and not progressing quickly so to give it a bit of momentum we decided to run a GEL WEEK

**‘How can we make a design system that saves time, is scalable, easy to use and to maintain?’**

GEL week was a whole week we dedicated to designing patterns and components

At this point the team collaboratively decided what the priorities were for the design system, how it would be structured and who would be responsible for what. This helped us ground that shared understanding and

made sure that everyone in the team was in the same place with the same goal in mind.

We also invited developers and product people to take part as some of the feedback from the existing gel work was that developers didn't always know where to get the right components

The goal was:

During GEL week the team decided on a structure which was pretty similar to the existing GEL structure

We agreed a structure of our design system:

Foundations (grid, typography, icons etc)

Components (there are fairly granular)

Search

Patterns (groups of components) Search,  
filter, list

And had lots of discussions about whether  
things are atoms or a molecules

And also agreed a structure for  
documenting components, which if we  
have all the parts we wanted would look  
like this

Overview

Working interactive component (this was  
one of the pieces of feedback from  
developers - with static components the  
they didn't know how interactions worked)

Links to demo



Link to code Github (meaning just a single repository for code as opposed to many)

Link to Sketch Library

Status indicator

How it works

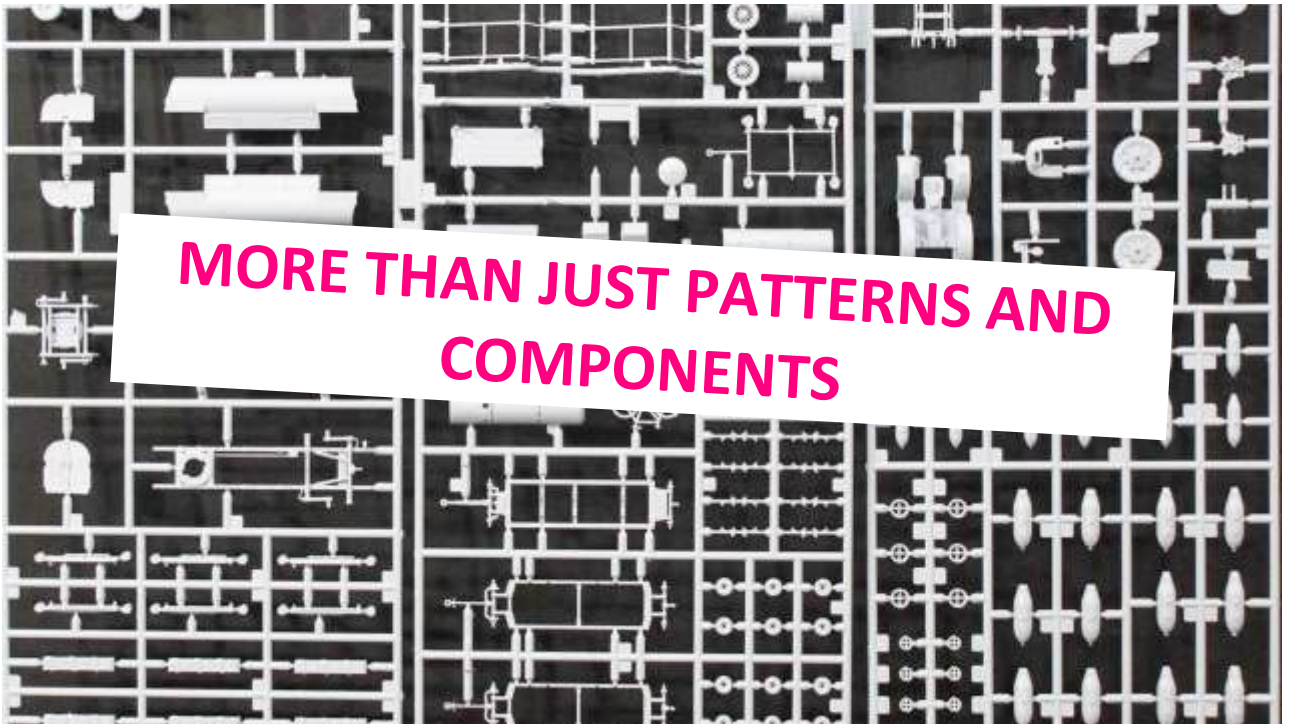
Rules

Variations

Pattern in action

Accessibility considerations

Research links



This develops things quite a bit further than the original gel work - it adds code components, examples of patterns in action and research links and addresses a lot of existing concerns.

All this give us good foundation to work from but we also knew from the start that when we talked about the

design system that it was not just about the patterns and components we used.

Internally we wanted to build up a toolkit of all the things we and the teams we work with need in order to design tools well.

So obviously we had:

Pattern library - shared centralised library of components,

But we also had groups working on

Language guidance - There are a couple of aspects of this

UI tone of voice and style

Depending on where the user is, we need

to strike the right balance between clear, instructional language and a friendly, welcoming tone.

In parts of a user interface where users are asked to do something, practicality trumps personality.

## Actions

There are lots of common actions people can take in our tools (save, add, create, publish to name a few) but these terms are often used differently in different tools and further adds to the feeling of inconsistency. So as well as having a consistent tone of voice having a glossary of actions helps us establish a shared vocabulary.

The next piece is the visual identity

Although our internal tools are brand is still important to the teams who produce them and is also a useful orienting indicator for users as our tools get more similar looking. Keeping some level of distinctiveness is important. We not need consistent just for consistency sake.

So this is all great - we have all this stuff that is useful!

So now we have a lot of this stuff we have to maintain it, from we know that having a design system is a living thing and that in order for it not to reach the end of its usefulness we know that we need to develop and maintain it and keep it as future proof as possible.

THE FUTURE!



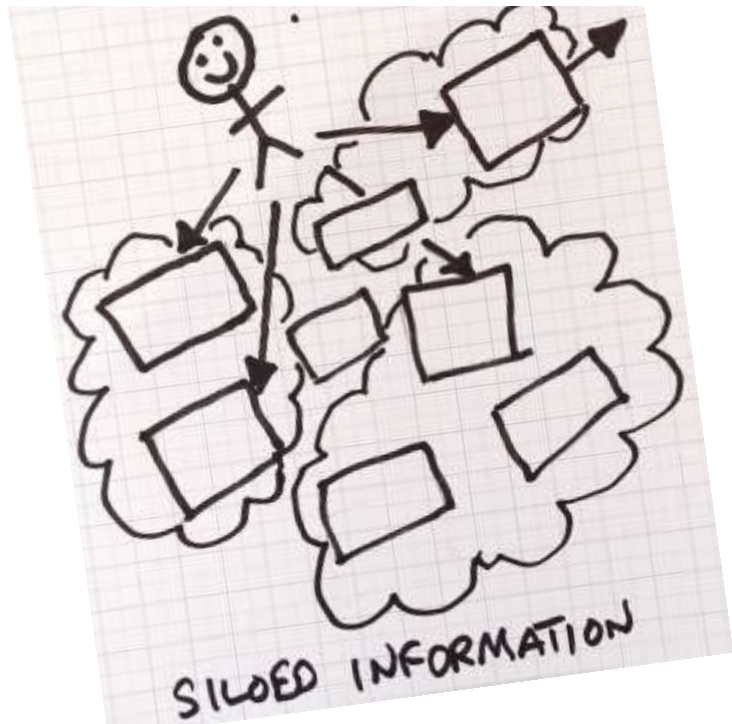
This means that while we are designing to make our day to day work more efficient we also have to keep an eye on what future developments might be coming and what opportunities we can identify and leverage to keep working towards our goal of more consistent and joined up systems

And this is where my job as an ux architect comes in - I'm using the term UXA interchangeably with information architect here but

I'm basically talking about the people whose role it is to pay particular attention to the organisation of information and data into places that users can understand.

In an organisation as large as the bbc, where you are constantly dealing with a huge number of moving parts and concurrent architectures it's important to have people looking at information architecture from a holistic point of view.

## FROM THIS



We know from our research that users find the multitude of tools and systems that they have use on a daily basis to them overwhelming, difficult to learn and time consuming to use. This is a problem that we know needs to be mitigated in any way possible.

In my role as an IA i tend to look at how we can move from having a lot of



different and separate tools and systems to having one coherent ecosystem with parts that people can use for specific things.

All this has to be managed while ensuring that users can still understand the spaces they use and navigate and complete tasks as they need to.

We basically want to facilitate moving from this to this so that users end up with a more coherent experience.

## TO THIS



We also know that we can't move to this future goal instantly, there are too many teams and products and stakeholders involved to coordinate completely, however as long as we know what the long term goal is, we can keep moving part after part into alignment with that.

Balancing current with future

Balancing detail with generality

Balancing distinctiveness with consistency

So achieving this is a constant process  
of

Balancing current need with future  
needs

Balancing a detailed view with high  
level

Balancing distinctiveness in specific  
places with general consistency

Ontology – define the things

Taxonomy – relationships between things

Choreography – organise the things

And information architecture can help us with this

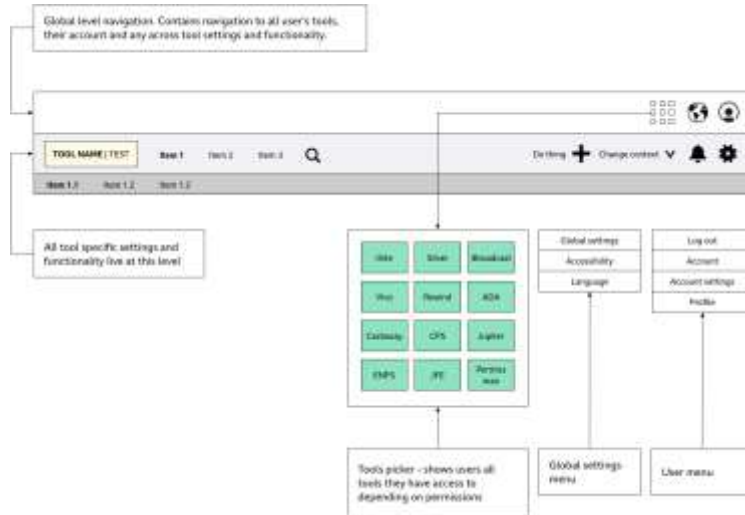
We can think of it as a process as consisting of 3 parts

Ontology - which is about defining the  
Taxonomy - the relationships between things

Choreography - how the things are

organised into a coherent understandable system

These three things can be used to facilitate consistency and understandability in any tool/service/product



So one of the pieces of work I've done in the past year focused on tools headers and navigation and how to standardise these are move towards a model where users could access any tool from one place. (Much like the model that Google use for their products)

Headers are a really good example of

a pattern that needs good information architecture in and this is also one of the first things we started developing into a design pattern.

There were a couple of drivers for this project other than the fact that we had a lot of different headers.

One was the fact that an organisation wide single sign on was being introduced and related to this there would be a standard way of granting access and permissions to tools ( as opposed to the permissions being managed by individual tools they would be managed in one place)

The second was the fact that development teams were wanting to share more of the

work they produced so some standardisation was useful.

So using that IA process we can start by defining what we have:

I started by auditing all our tools headers

We had literally a million

And they were all different

And there were some really strange things in them

So what I ended up with was a huge list of things that could be in headers many of which were called different things but were performing the same function.



By sensechecking this list with people working on each of the individual tools I managed to create a list of 22 needs that the header might have to fulfil and a standard name for what that thing was called. I also categorised the items into groups to make sure anyone who used the pattern would know what purpose an item was serving, eg. branding, navigation, action, user account, settings.

## Taxonomy

### Refining rules for the thing working

The second part of the process was working out the rules for how each of these 22 things could be used and where.

Some things were mandatory (eg, user account)

Some items had provide for multiple needs (eg, logo performs the function of navigation and branding)

Some had to be flexible enough to accommodate the needs of different tools (eg, quick action, in one tools it might be 'new item' in another it might be 'upload')

Some things might only exist in the as cross tool items in future.

So once we had these items and the rules for their use we knew what the basis for the header pattern was.

So, so far so straightforward..



## DESIGNING UNDERSTANDABLE ECOSYSTEMS

The next part of that IA process is about arranging the things you have in to a coherent and resilient whole and just having a pattern isn't enough to do this – people (both other designers and product teams) need to have visibility of how that patten fits into the existing architectural ecosystems and how it may develop in future

At the point I started working on the header, both the standard permissions project and the component sharing project were purely conceptual - there were no UIs available to show what these things might mean or what impact they might have and in some ways they were still fuzzy and not completely defined – this is a point where I often end up finding myself working as an IA

I think about this part as making things tangible (whether that's data models, interactions between systems or flows metadata) and communicating these things to help form a shared vision.



impact different tools. This helped visualise things for both the team developing things and highlight any oddities this might end up producing in the tools. It can also help feed back into any points where concepts are ill defined

So a lot of this choreography part is about visualising and showing how lots of things might come together to become a consistent whole and about familiarising people with ideas and helping them see where they might create problems, but also where they create value and possible future opportunities.

Having this overview gives space to the patterns and design and helps us design to a common future goal.

And all of this together provides a stable information architecture to design a patterns from .



**Know what your goal is**

**Build strong foundations to achieve that goal**

**Work together towards it**

So in conclusion

It's not just about having a design system, design systems are living things that need to constantly adapt to fit the changing needs of an organisation.

And keeping a design system alive means:

Knowing what your goal is

Building in strong foundations to achieve  
that goal

Working together towards it

**Thanks!**