NEXUS LABS

NEXUS LORE

The Smart Building Blocks

THE PAST, PRESENT, & FUTURE OF THE SMART BUILDINGS INDUSTRY

2022 EDITION

WRITTEN BY:

James Dice, PE, CEM, CMVP is the founder of Nexus Labs and has over 12 years of experience in the North American Smart Building Industry.

Nexus Labs is an online community, school, and consultancy striving to create smarter, more efficient buildings. Starting in 2019, the Nexus Community has grown rapidly and sees new members each month from across the industry. With regular gatherings and daily online discussions, players from organizations all around the world are working together towards digital, decarbonized buildings.

Nexus Lore: The Smart Building Blocks

The core concepts of the Nexus newsletter, the Nexus podcast, in the Nexus Foundations course, in Nexus Pro gatherings, and in the community chatroom

Table of Contents

Introduction

The Oxford Dictionary defines 'Lore' as:

*"**A body of traditions and knowledge** on a subject or held by a **particular group**, typically passed from person to person by word of mouth."*

We, the Nexus community, are a ***particular group***. That might be the understatement of the year...

And we've been building up a ***body of traditions and knowledge*** on the subject of smart building technology for the last 2+ years. I know this because our (particularly) nerdy terms and acronyms—IDL, ASC, etc—are gaining steam and showing up in specifications, investor pitch decks, and conference agendas around the world.

Part of my job as the founder of this amazing community is to help define and curate our lore. So today, we're kicking off a ten-part series of newsletters on the core concepts that come up again and again in this newsletter, on the Nexus podcast, in the Nexus Foundations course, and in Nexus Pro gatherings, and in the community chatroom.

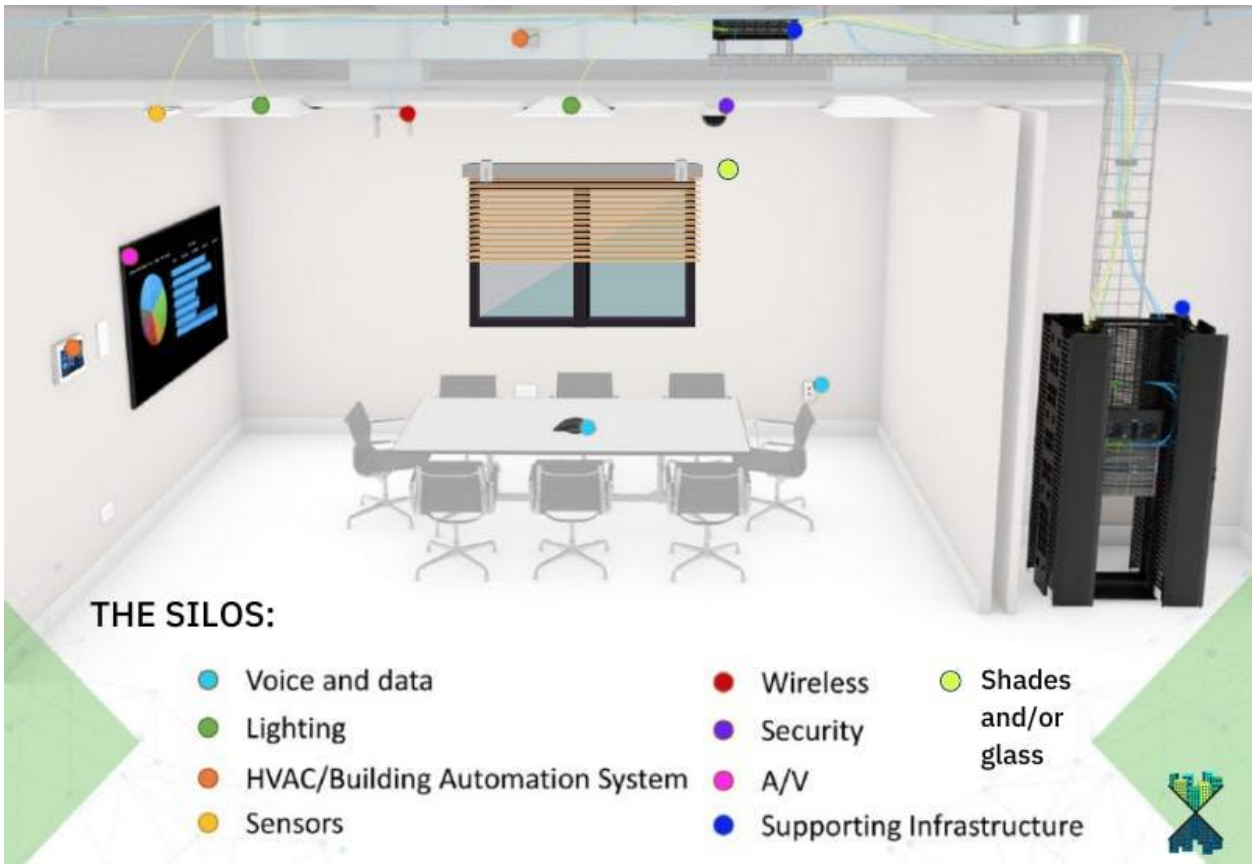
We'll call them The (Smart) Building Blocks. As always, we'd love your feedback.... Lore is never written by one person.

We'll start at the beginning...

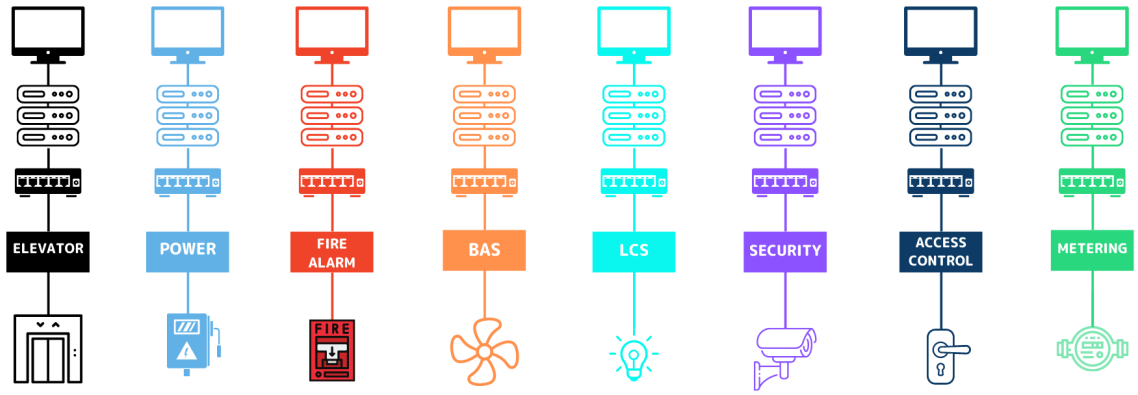
The Silos

Picture a single room: perhaps a conference room in an office.

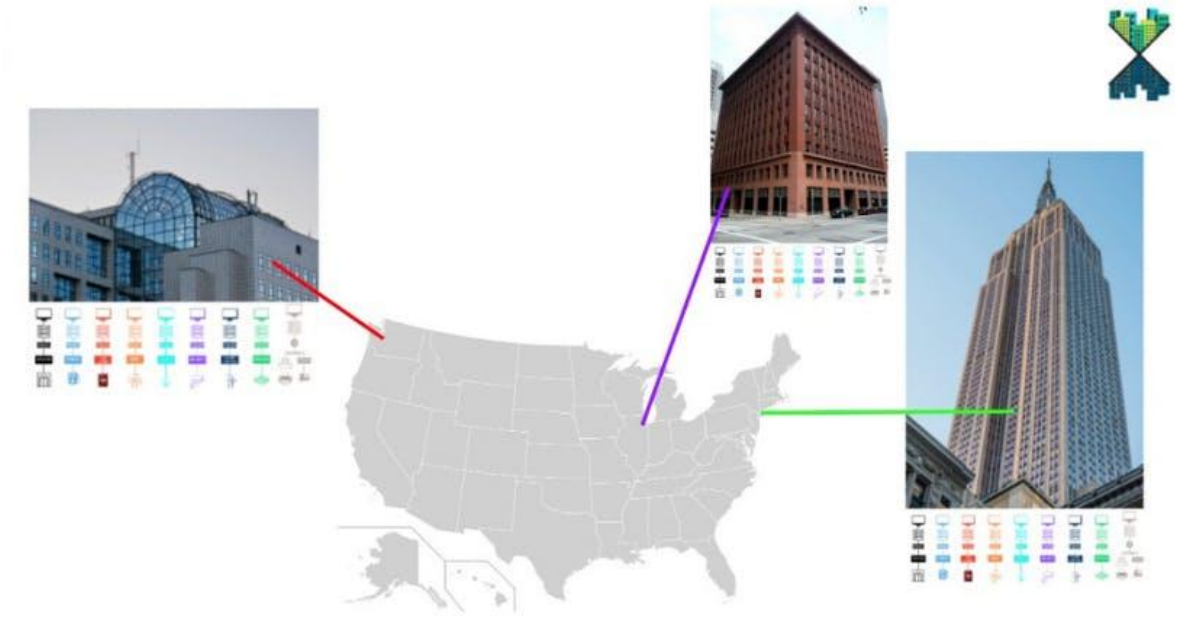
You have all these siloed systems:



And that's just one room! In a building or campus, you have full tech stacks for each system serving all the other rooms.



Now picture a portfolio of many buildings! Different silos for each one.



These systems exist to provide the core outcomes building owners need:

- A productive, safe, healthy, efficient, comfortable, collaborative occupant experience
- Minimal carbon emissions
- Minimal expenses, maximum revenue

But right now, in most buildings new and old, these systems could do a much better job at achieving those outcomes. There's room for improvement in (at least) four ways.

First, each silo can be further optimized to play its role better. For example, the HVAC system could get better at maintaining comfortable conditions.

Second, silos could share data to aid in that optimization. For example, the sensor silo could share occupancy data with the HVAC system, which could reduce energy during unoccupied times.

Often, the outcomes are competing with each other. For example, imagine you're asking the occupant of the above conference room to control the blinds, lights, or thermostat... they're not productive, they're probably not doing a good job, and they're probably annoyed.

That brings us to our third need for improvement: We need to optimize all the systems in unison to make sure we're not jeopardizing one outcome in favor of another.

Finally, our fourth opportunity: our silos are typically designed with only one type of stakeholder in mind. Buildings are diverse ecosystems of stakeholders—each want different things.

For example, an HVAC technician wants to fix that damper actuator, the property manager wants to minimize hot/cold calls and keep tenants happy, the sustainability manager wants to know how much energy the actuator fix will save. How can one user interface help all of them?

With so much room for improvement, why do silos exist? While many in our community are doing great work removing silos in the construction process, the reality is that these silos are baked into how buildings are built.

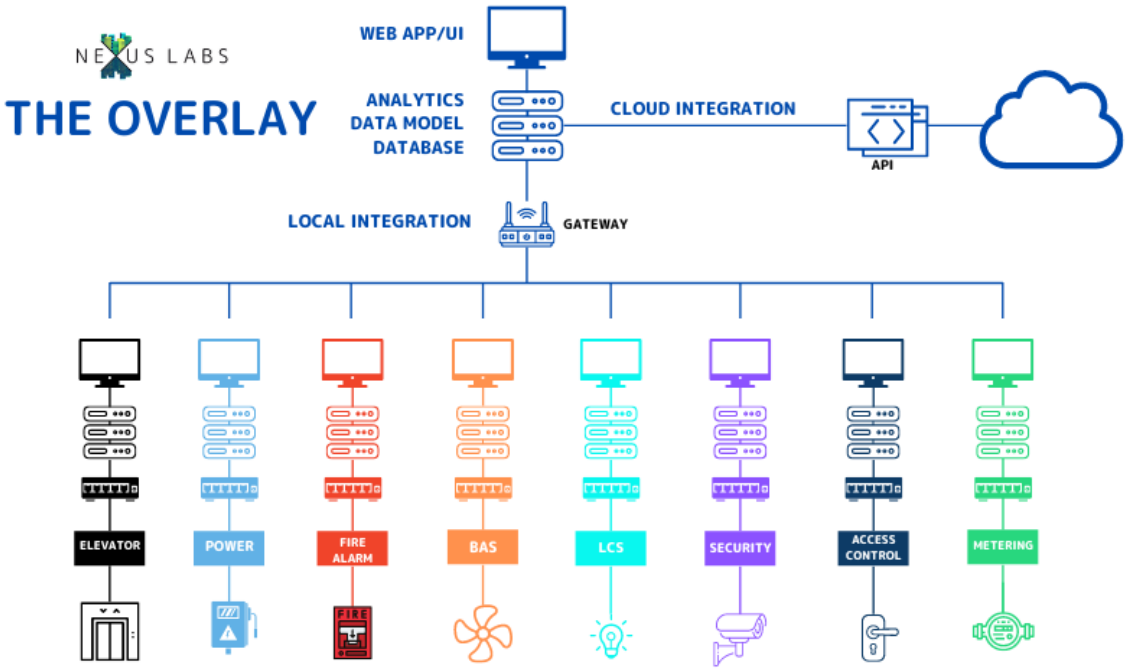
Most of our existing buildings were constructed with separate specifications, contractors, technology stacks, and service vendors for all the different systems in the building. And after the building gets built, those silos tend to last a long time!

The Overlays

There's a wide (and growing) gap between what building owners need and what siloed systems are capable of.

While our siloed stacks are getting smarter (more on that soon), we can't just rip and replace everything in our buildings with the latest and greatest. Practically speaking, smart building solutions need to solve the silo problem by integrating with existing systems in the portfolio.

We accomplish this using “overlay” software. An overlay by definition will “sit on top of” existing systems or applications to allow access to data and functionality from the underlying systems.



I think of this architecture as phase 1 of the smart buildings industry. A vendor installs an integration device on the local network, pushes data to their cloud, organizes it according to their standards, does some fancy math and visualizations, and serves up a software application.

And these solutions are downright amazing compared to what we had before (and what many building owners still have today). For example:

- Overlays like Clockworks Analytics can diagnose issues the HVAC systems in almost real time.
- Overlays like Bractlet can create a physics-based energy model, calibrate it to a building's actual performance, and create scenarios for decarbonization.
- Overlays like BrainBox AI can control HVAC better than the control system can.
- Overlays like Comfy can give a building occupant the local temperature control they never had and allow them to provide valuable feedback to the building operator.

All of those are incredible innovations! But... here's why phase 1 is coming to an end:

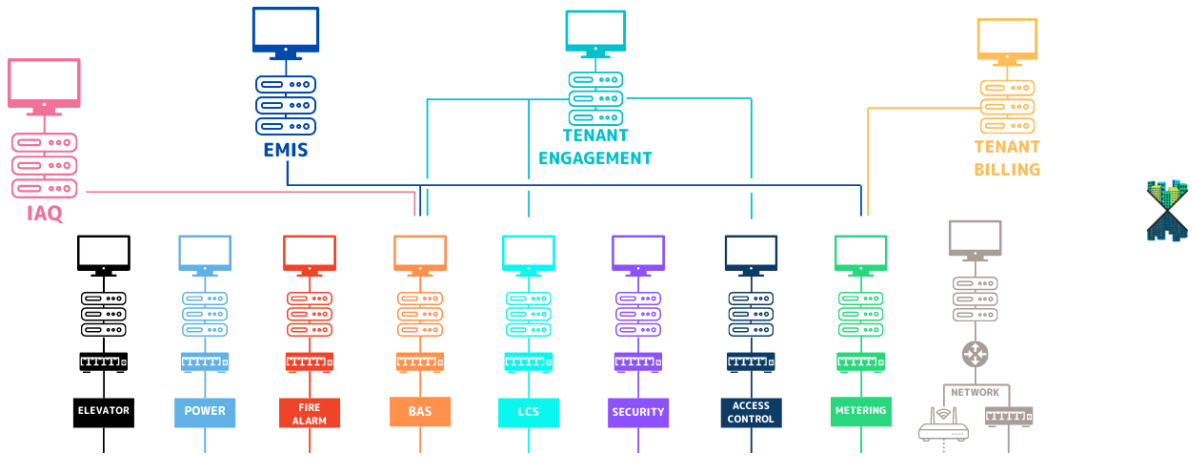
Solving individual problems with separate overlays creates new silos.

This phenomenon is common across all industries that are in the midst of digitization. Single-purpose overlays are widely termed "point solutions", which PC magazine defines as:

> *Solving one particular problem without regard to related issues.*

Point solutions solve one or two problems for one or two stakeholders, leaving out adjacent workflows, problems, and jobs to be done.

Building owners are left with point solution spaghetti, like the 53 building portfolio with 50 different technology providers. With so many balls to juggle, O&M stakeholders find their core daily workflows to be inefficient, confusing, full of task switching, and difficult to keep up to date. Managing this complexity takes away from, rather than enhances, the ability to operate and maintain their buildings.



Building owners pay for multiple point solution vendors to do the same work at each infrastructure layer of the stack—resulting in redundant integration, data storage, and data modeling layers that don't easily communicate outside of each vendor's stack.

Phase 1 isn't ideal for vendors either. They need to build that infrastructure layer on every new project, delaying time to value, lengthening sales cycles, and increasing customer acquisition costs.

Point solutions are a stop on our journey, but they're not the destination.

The Horizontal Architecture

> *"I see a lot of people buying apps before they've bought the smartphone. (Our current smart building strategy) is really about building up that smartphone foundation."*
> *—Jon Clarke, Dexus, on episode 62 of the Nexus podcast*

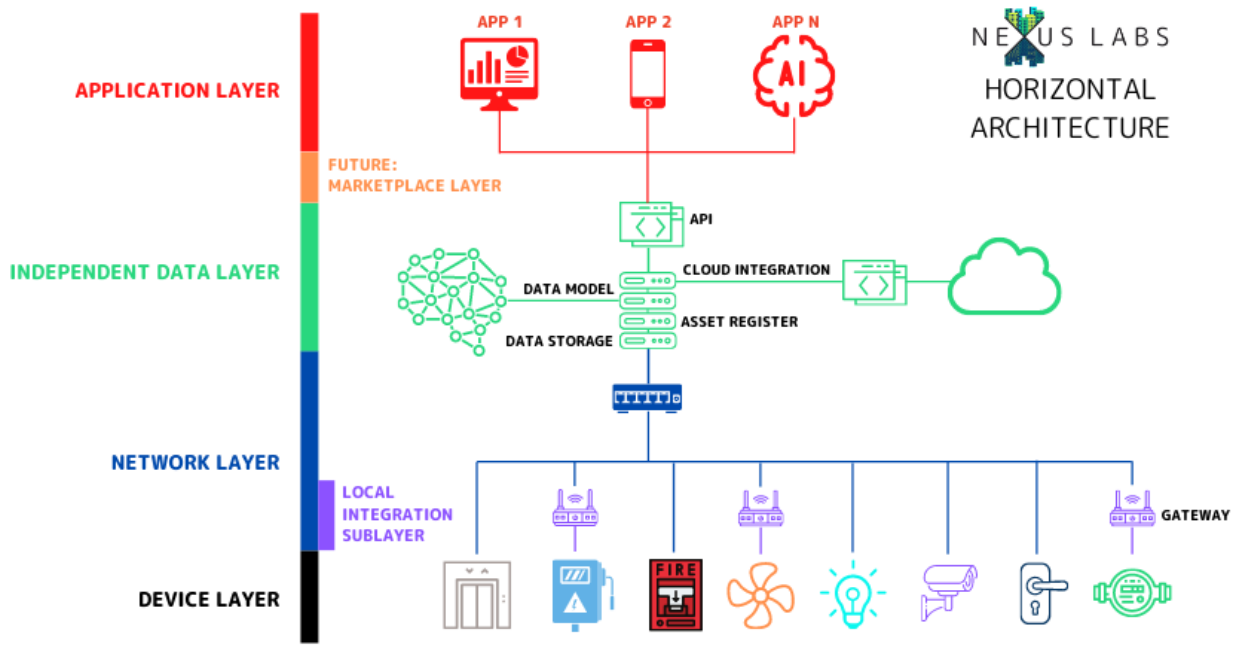
Point solutions are full vertical stacks, but what makes each valuable is the application layer, not the rest of the stack. If point solutions represent phase 1 of the smart buildings industry, phase 2 is the response: the horizontal architecture.

Building owners are realizing they don't want to buy redundant layers from different application vendors, they want any application they buy (now and in the future) to share common infrastructure.

Another way to think about this phase change: we're moving from a smart buildings project mindset to a smart buildings program mindset. Projects are single-focused, time-bound, and implemented in silos. Programs are comprehensive, ongoing, and integrated into core operating procedures—and the architecture must match this new approach.

So what are the horizontal layers? There isn't one standard yet. The digital buildings team at Google created the Building Operating System concept. The crew at Monday Live! created The Smarter Stack framework. I'm sure there are many more out there.

Looking at these, plus all our podcast interviews and approaches I've heard about, here's my best crack at synthesizing everything I've learned on this.



Starting at the bottom, we have horizontal layers (devices, converged network, and independent data layer) that enable bi-directional communication with overlay software applications. Portfolio level standards (e.g. cybersecurity) determine how each layer is set up, how each layer is maintained, and how new devices are introduced.

To me, the nerdy details of each layer are fun to explore (and we will over the next few weeks), but the most interesting question is: what does our agreement on this new approach mean for the future?

Imagine a future where every building's Devices, Network, and Data Layers are set up to scalably, securely, and reliably enable any smart building application. If that were true, ALL of the outcomes we're collectively working towards will be easier.

That doesn't mean we should halt all progress with deploying new cutting edge applications. But pursuing applications without acknowledging the importance of the infrastructure needs to be an artifact of the past.

The Device Layer

How to explain the Device Layer—the bottom layer in our Horizontal Architecture?

First, we must start with the elephant in the room. As covered in our whitepaper, 87% of commercial buildings don't have any digital systems. That means we still need to install a lot more digital devices to enable everything higher up the stack.

For the remaining 13%, which tend to be our biggest, richest, and most complex buildings, the device layer can be summarized like this...

Siloed systems—each with their unique purpose, inputs & outputs, key stakeholders, and life safety functions. Each with their separate stacks, including networking, asset registers, data models, data storage, user interface, and devices.

When you log into these systems' software applications, you feel like you're logging into the 90's. That's because you are—these products haven't changed much since then.

And neither have the business models involved. To avoid the commoditization of their hardware and to fuel lucrative service contracts, providers of siloed systems attempt to preserve their ownership over the full stack by locking others out and making half-assed attempts at the interoperability standards the industry needs.

Matt Schwartz said it well: owners everywhere are forced into navigating complex upgrades, unlocking from proprietary systems, spending large capital for simple changes, and simply trying to make sense of this siloed world.

Finally, now that we've grounded the discussion in what you'll still find in most buildings, let's think about all the changes happening. The following are the main themes I see. They're not evenly distributed, but they're disrupting the device layer status quo building-by-building.

🔗 Trend #1: Everything will have an IP address

There's two layers to this one. First, ethernet is replacing the serial networks of yesterday, meaning the number of traditional devices on building IP networks is exploding, from thermostats to locks to elevators. Second, many more types of devices are now getting IP addresses, from Peloton bikes to coffee makers to digital signage to trash cans to refrigerators to windows to vehicles to PV inverters.

📊 Trend #2: IoT sensors create valuable new data

We're seeing entirely new (and yes, siloed) IoT sensor stacks. These are coming because of Moore's Law—with sensors and gateways becoming super cheap, you're seeing a lot of new players entering the market. Examples include people counting, IAQ, kitchen equipment monitoring, submeters strewn everywhere, leak detection systems, and seismic detection. This data enables new use cases for the layers up above.

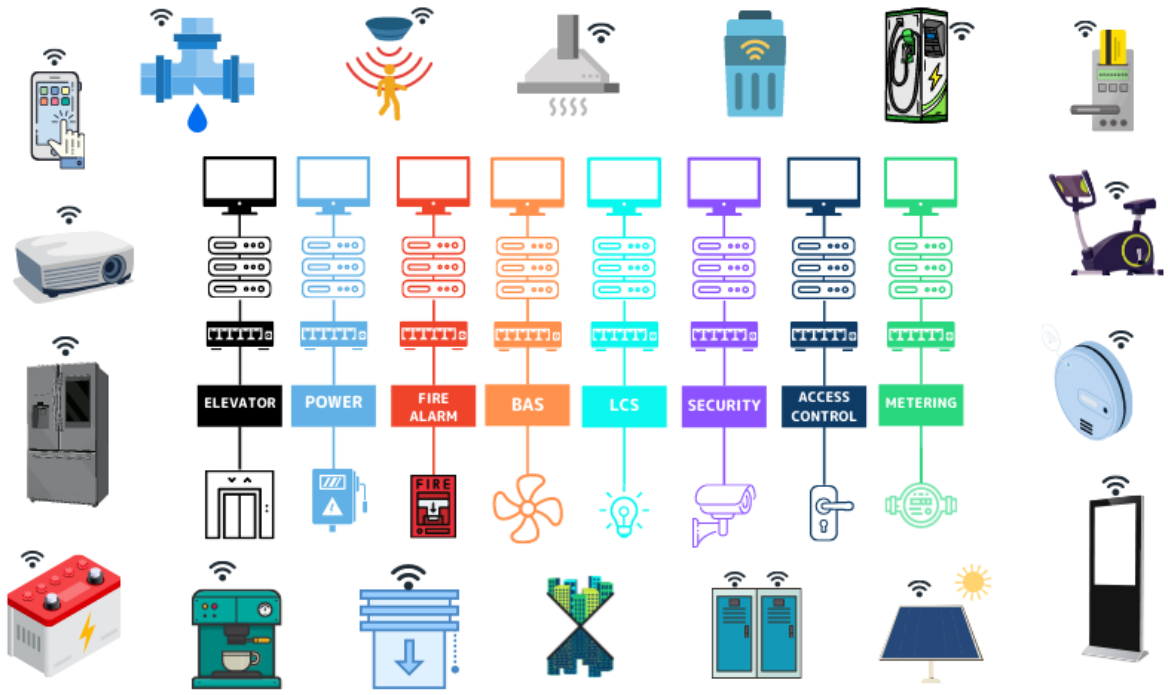
💡 Trend #3: Traditional devices are getting smarter

What if we built traditional controllers with modern hardware and software technology? It's happening already: look at Passive Logic's HVAC controls, Span's electrical panels, and Belimo's smart valves as examples. Whereas you used to simply connect a Belimo valve to the BAS, now you have a valve that can do a ton of stuff on its own by connecting to the network and the cloud—such as analytics, control, software updates, and remote support.

📱 Trend #4: Devices to engage with the occupant

Finally, the device layer now includes occupants' personal devices. As an example, commercial office is rapidly changing and becoming more focused on providing a great tenant experience. This is increasingly done with a mobile app, but can be delivered using other devices too. Examples include IAQ data being displayed on a kiosk, room booking devices, point of sale devices in the cafe or kitchen, connected kitchen/coffee appliances, lobby directories, etc.

To summarize: It's not just our traditional silos. We have all these new devices. All these new connections. All these batteries, power supplies, and other dependencies. It's a bit of a mess, right?



As a result, building owners are left with (at least) three big questions:

- How do we manage all of this complexity? *E.g. What devices do I have? Are they working? What are they measuring? Are they all updated?*
- How do we secure our networks with all of these new connections?
- How can we simplify how this layer is implemented across the portfolio?

In order to enable the outcomes we talked about in Part 1, owners are left with a double edged sword: they must abstract away the complexity at the edge but also fully respect it and treat it with the detail it deserves.

The Network Layer

We've unpacked all the devices that now need a network connection. That's the next layer in the stack: The Network Layer. In order to enable use cases for technology, this layer needs to:

- allow remote access for administration;
- get data to the cloud;
- allow communication from the cloud to the devices; and
- allow communication from device to device.

Pretty simple, right?

Not so fast. As most of us have experienced on our projects over the years, if we don't treat the network layer seriously, things can blow up rather quickly. Network problems hold the industry

back by making smart buildings look risky, delaying project success, increasing costs, and making stakeholders question whether the juice is worth the squeeze.

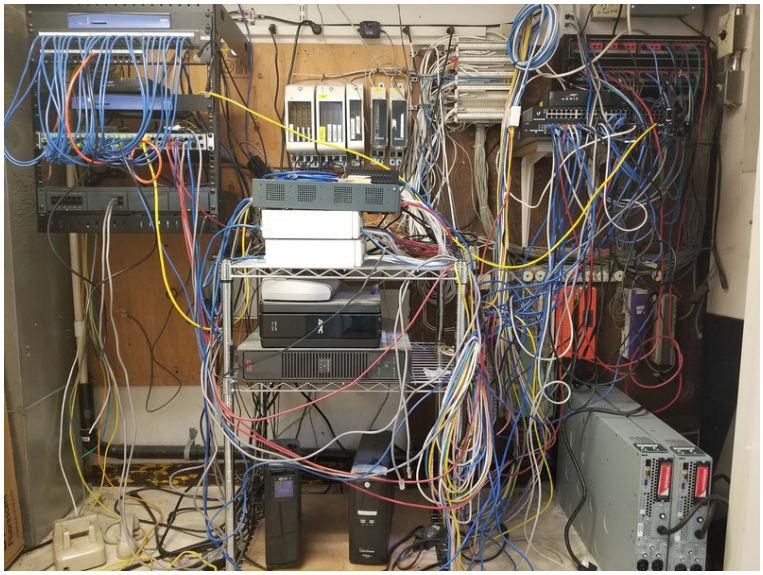
What problems are we talking about? Let's name a few...

🤫 When the IT folks close your hole in the firewall because they didn't know what it was.

😬 When you're scanning the network for the devices you know exist but are nowhere to be found.

🥴 When the power goes out and someone needs to reboot every device manually, but they forget the penthouse where all the air handling units are.

🤦 When you're on a new construction project and there's no budget for the full commissioning scope, and yet three different contractors are running redundant cabling.

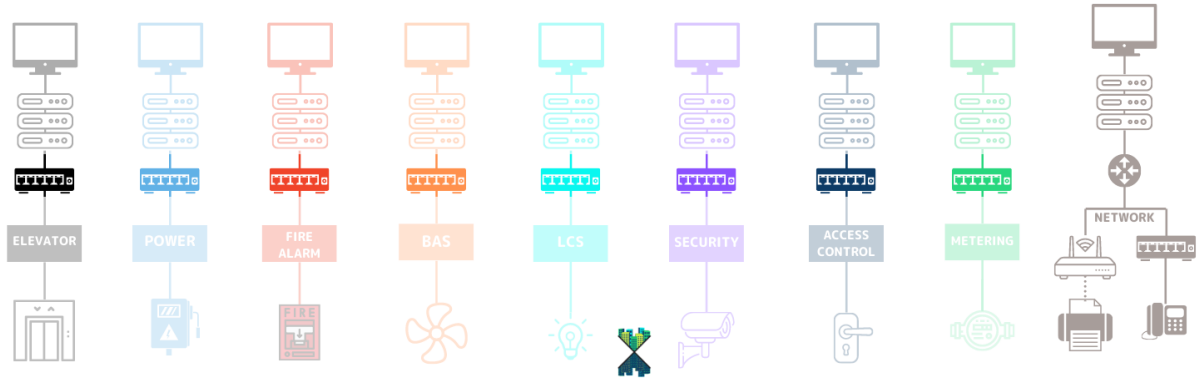


😳 When *that's* where you're supposed to plug in your device.

😭 When the demand management project fails because the internet went down on the hottest day of the year and your software couldn't send the command to curtail the load.

😤 When your access control system is hacked and held for ransom.

You get the idea... All of those problems, and more, are solved at the network layer of the stack.



A dedicated layer with its own hardware, software, standard operating procedures, and key stakeholders who take responsibility for doing it right. It should be converged, monitored, maintained, and have redundancy.

After all my interviews so far on the Nexus Podcast, I've learned that the key word is "responsibility". We have this IT network or networks, some of which have smart building functions and importance. IT teams generally have no understanding of or experience with networked building system protocols and communication methods. And we have this OT network or networks, that's in varying degrees of convergence onto a common internet protocol infrastructure. OT teams, meaning Engineering or FM or their vendors, have experience with BAS and related systems, but little understanding of/experience with enterprise networks and IoT.

Someone must take the responsibility to fill the IT/OT gap to enable the success we're looking for, as a mutual understanding of both IT and OT systems is needed to manage the smart building network effectively.

The typical building has a hodgepodge of unmanaged siloed networks, "flat" networks running multiple building systems through one switch, building systems connected to the IT network, and building systems wide open on the internet.

So who *should* take responsibility? I think the best solution depends on what organization you're talking about. In some organizations, like Google, the IT folks take full responsibility. In other organizations, the IT team simply doesn't have the bandwidth or skill—they've found success leaning on dedicated service providers. Finally, I think most of the debate on this topic resides around whether OT vendors should take responsibility (a topic for another day).

What's not up for debate? The ways the current network hodgepodge causes dumb buildings. The enabling infrastructure that lets us fulfill our vision for smarter buildings.

The Independent Data Layer

Since we're past halfway, let's start with a quick review...

In short, we need more out of our buildings, but siloed systems hold us back. In an attempt to improve outcomes, the number of connected devices and point solutions has exploded, the silo problem is only getting worse.

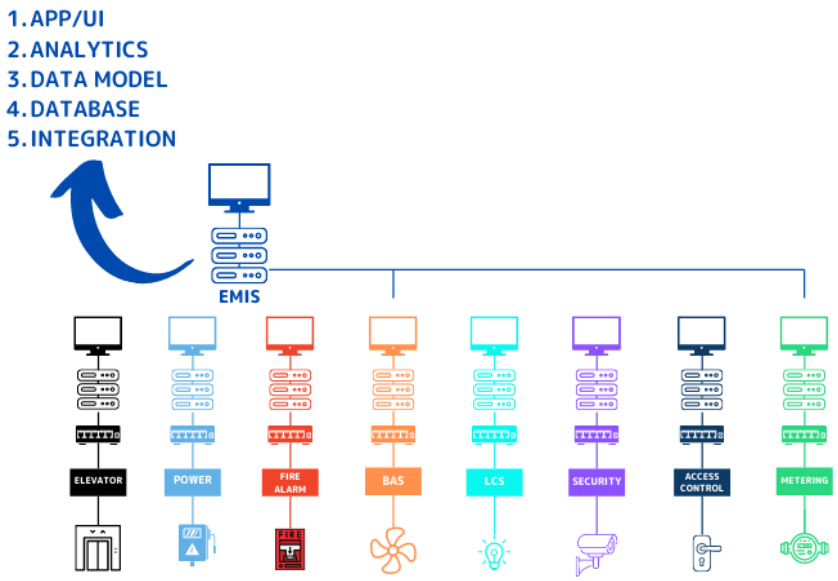
The horizontal architecture lights our path forward by focusing on building a sturdy infrastructure that enables smart building apps, starting with the network layer.

Next up in the stack: The Independent Data Layer. It's a continuation of that horizontal philosophy: providing the infrastructure for any smart building application that is *independent* from each of them. Why independent?

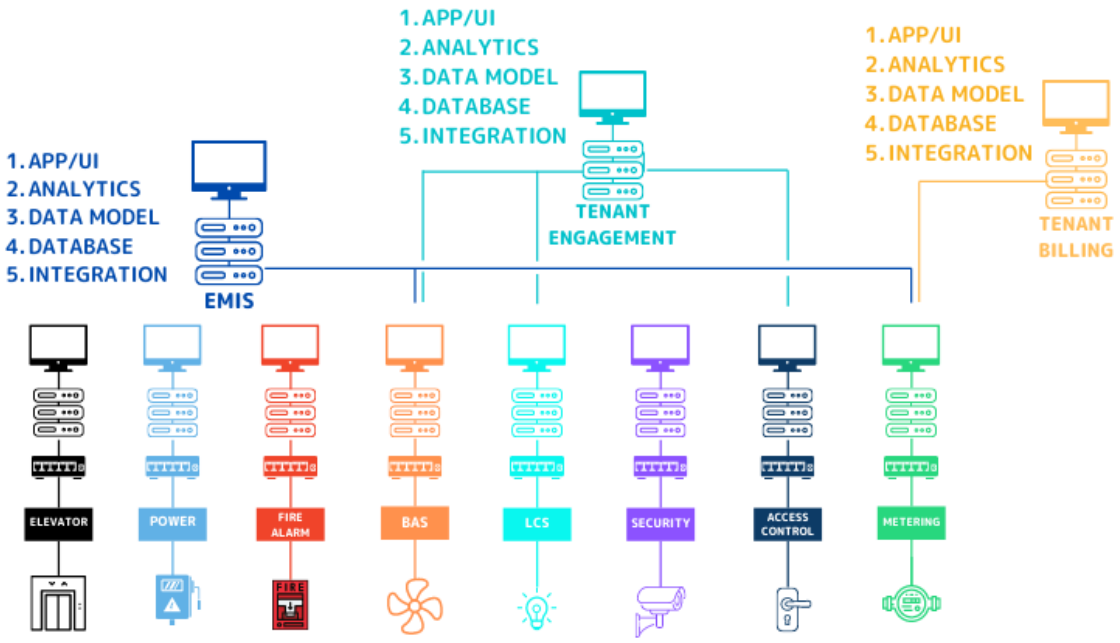
In short, because integration is hard.

Any time we want to deploy a new tech use case, we need to integrate with existing siloed systems to communicate with them. That integration is labor-intensive and often takes much longer than it should.

Each overlay solution provider sends their integration team on-site and deploys a full-stack overlay solution—complete with integration hardware/software, a historian database, a mostly-custom data model, some sort of analytics or data transformation, and a user application—on top of the existing silos.



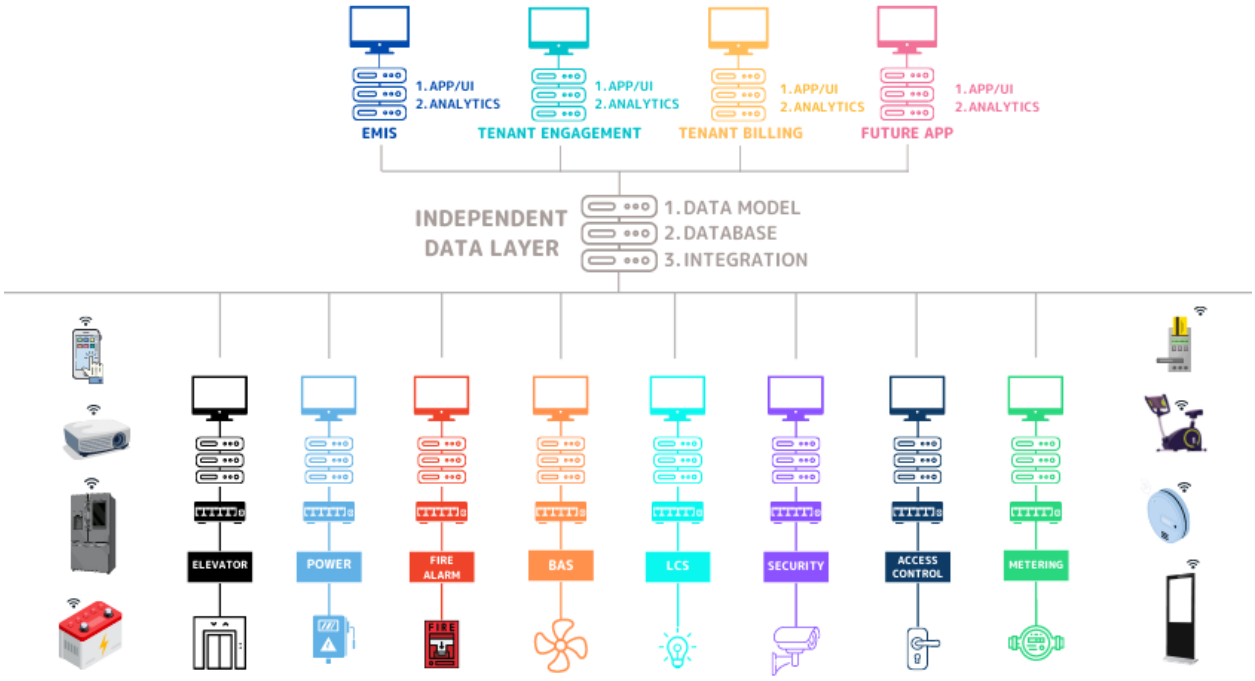
And then when we want to deploy another use case, we deploy another full-stack overlay, sending the next company's integration engineers on-site to do their thing.



Now we're doing some cool stuff with data and providing an app to occupants, but we've also created new silos. If we need to connect these new silos, we're then creating new point-to-point integrations that make it difficult to keep the whole thing updated—there's no single source of truth on what the data is and what it means in context.

Finally, we've created an architecture that's difficult to adjust and adapt. For example, if I want to remove a vendor and pick a better one, I need to literally start over from scratch for that use case. If I want to pilot multiple vendors and compare them to each other, each will be doing redundant work.

The core promise of the IDL is to minimize all that integration work by abstracting away all the complexity of each silo, while chopping up all those overlaid stacks into centralized infrastructure.

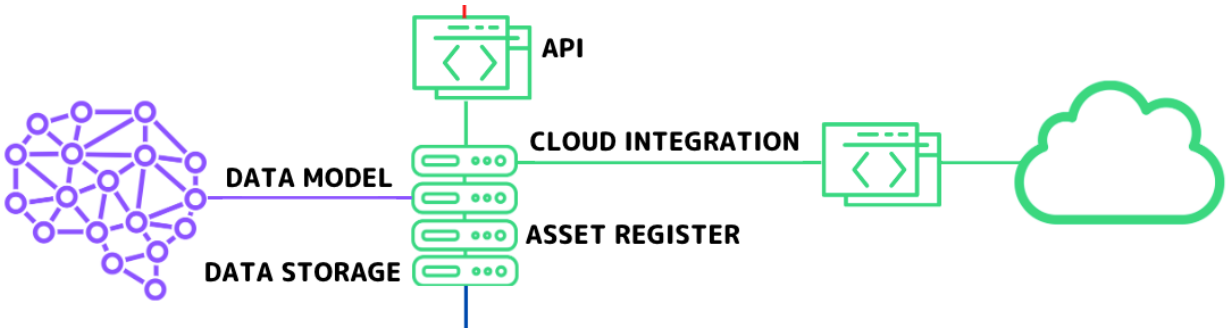


The resulting value proposition is reduced deployment costs, reduced time to value, future flexibility, and choice at the application layer.

> *I think that's ultimately the problem that an independent data layer solves for is giving the owner, the operator, the facility manager choice on what kinds of solutions they want to bring to bear for different kinds of problems within the system."*
> —*Andrew Rodgers*, co-founder of Ace IoT

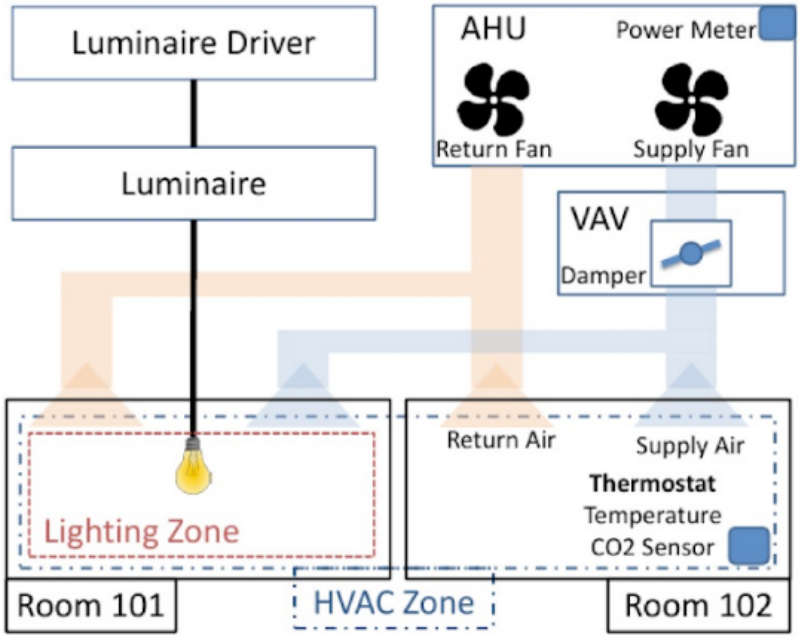
The Data Model

Let's pause our exploration of each horizontal layer and zoom in on one key part of the Independent Data Layer: the Data Model.



In order for our applications (the next layer in the stack) to do their thing, they're going to need *context*.

The applications need a way to understand all the data they're consuming, including all the underlying devices and how they fit together into a system of systems. As an example, the diagram below shows just two siloed systems (HVAC and lighting) and just two rooms of a building.



Source: Brick Consortium

See how all these devices, rooms, zones, and systems overlap with each other? Both the lighting circuit and the AHU and the VAV all “serve” room 101. The AHU and the VAV also serve room 102, which has the thermostat and the CO2 sensor that control the operation of the system.

Any application that uses HVAC data will need to know these facts—so how do we communicate those concepts in a standardized way? In most buildings today, we don't. All the silos and teams of humans represent these concepts differently.

Take naming for example: The architect might call it Room 101; it’s in the CMMS as Room_101; the BAS doesn’t even have a concept of rooms; the lighting controller calls it Zone-1 (and every controller has a Zone-1); the ACS calls it a 'Conf room'; the tenant calls it the Zeus conference room because they’ve named all their rooms after Greek gods. Crazy, right?! And naming is just one aspect of data modeling!

The key detail here is machine-readability. Most of the time we humans can infer that it's the same room, but a computer would see them as distinct concepts unless programmed otherwise.

The migration to more interoperable systems comes with more machine-to-machine communication—which requires data to be understood by a computer natively. If a common data model doesn't exist, then us humans need to map disparate data models to each other for two systems to work together.

And trust me (since I've done it): **mapping is time-consuming, mind-numbing, and costly.**

This is where standards come in. Using a standard ontology—like Project Haystack, Brick Schema, or Google's Digital Buildings Ontology—minimizes mapping and ensures the data model is ready for any application. However, although Project Haystack (the original) has been around for over 10 years, we're still in the early days of our industry's data modeling standards development process.

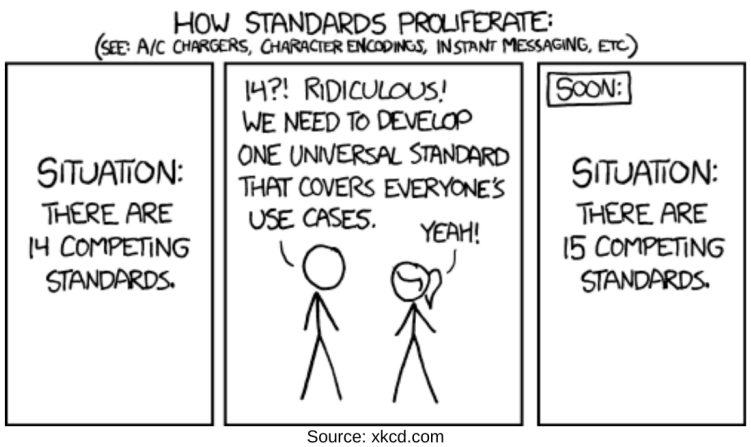
Here's a quick snapshot of where we're at today:

- **There are many standards**—Haystack came first, and then Brick, and then Google added a third. The Microsoft ecosystem also added their DTDL, which other efforts have extended to cover real estate (E.g. RealEstateCore). The Passive Logic team is now creating Quantum.
- **They don't yet cover all silos and use cases**—Haystack was created mostly for HVAC and metering. Brick and Google cover other silos, but there isn't one standard that covers all smart buildings use cases.
- **Most vendors have created their own**—Naturally, each vendor creates their own standard to save their team time, enable their own use cases, and develop their products. Some are extensions of industry standards, most are not.
- **There are many efforts at convergence and/or alignment**—Efforts like ASHRAE's proposed standard 223, the DOE's BENEFIT project, and the Ontology Alignment Project are hoping to create commonality.
- **Convergence is difficult, slow, and mostly unpaid work**—This topic is a really deep rabbit hole and lots of people get lost in it, including me! The work of convergence/alignment is mostly volunteer... and it ain't fun.

Given that, should each building owner just give up and do their own thing? I don't think so...

While the status quo is messy, the north star is to start with an existing standard and continue to adapt your data modeling methods (and your IDL) as the standards evolve.

That way, you're not expecting the rest of the industry to adapt to your new standard…

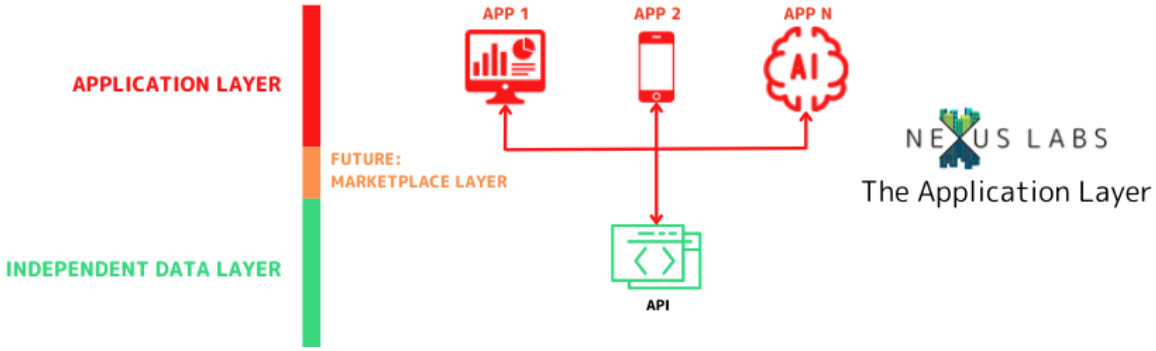


Source: xkcd.com

The Application Layer

In each layer we've discussed already (device, network, data layers), we've been talking about enabling infrastructure. In most buildings today, this infrastructure is brittle, unreliable, and not secure.

However, assuming we've solved all those layers, we're now ready for the applications the infrastructure enables. The Application Layer sits “on top” of the IDL and provides outcomes to users through mobile apps, web apps, or process-based applications.



Have you heard the term 'killer app' before?

It's defined as follows: *a computer application of such great value or popularity that it assures the success of the technology with which it is associated.*

In 1979, the launch of the first killer app, VisiCalc, ignited widespread business and personal use by consumers—which couldn’t have been imagined decades earlier when computers were first developed.

Here's a fact to ponder: There are no apps that *'assure the success'* of smart buildings technology. While there are plenty of examples of the cream-of-the-crop software providers, there is no killer app that is tearing through the real estate world.

Instead, there are hundreds of application vendors with traction in hundreds or maybe thousands of buildings. Remember: there are ~6 million commercial buildings in the US alone...

What's this mean for this layer and the industry as a whole?

First, it means there's a lot of opportunity—and that's exciting! Second, with all these distinct apps, it means that interoperability problems between applications are a growing concern (that's where the IDL comes back into the picture). Third, it means there's bound to be consolidation through mergers and acquisitions—and it's already happening.

Finally, it means the marketplace is confusing—some call it the Startup Swamp. There are over 400 application vendors on The Nexus Vendor Landscape! In our Foundations course, we teach students a framework that helps wade through that muck.

Because our industry's buzzwords don't hold much weight, we must first think about applications in terms of the *Capabilities* they provide and the *Use Cases* they enable. Apps in our industry have 5 general types of capabilities:

1. Centralize/visualize data
2. Analyze data to produce an insight
3. Control underlying devices and systems
4. Optimize human workflows
5. Engage stakeholders

With that basic understanding, we can then narrow the list from 400 down to a shortlist to start taking seriously.

The building owners I've spoken with are implementing several strategies as a response to the messiness: do pilot projects to learn more, hire internal smart buildings experts and external consultants to sift through the Startup Swamp, wait it out, remain flexible, and focus on infrastructure first.

The leaders have done all of the above and now they're working on portfolio-wide rollouts of applications that enable their top priority use cases, which vary by building type. For example, one large office building owner told me they're rolling out applications to:

- automate energy and ESG data collection & reporting;
- digitize occupant amenities; and
- digitize commissioning and maintenance (E.g. fault detection & diagnostics).

For the others, at some point, they must take action whether it's messy or not.

The Apps We Need

There are endless application providers in the Startup Swamp. There's also endless hype around the mythical "single pane of glass", the "digital twin", the "building operating system", and the "smart building platform".

And yet, having seen it all, there's still room for improvement in how well today's applications meet building owners' true needs.

The transition to a horizontal architecture will (theoretically) set application providers free. They can stop focusing on deploying infrastructure layers, then repurpose all that time and resources towards building better software.

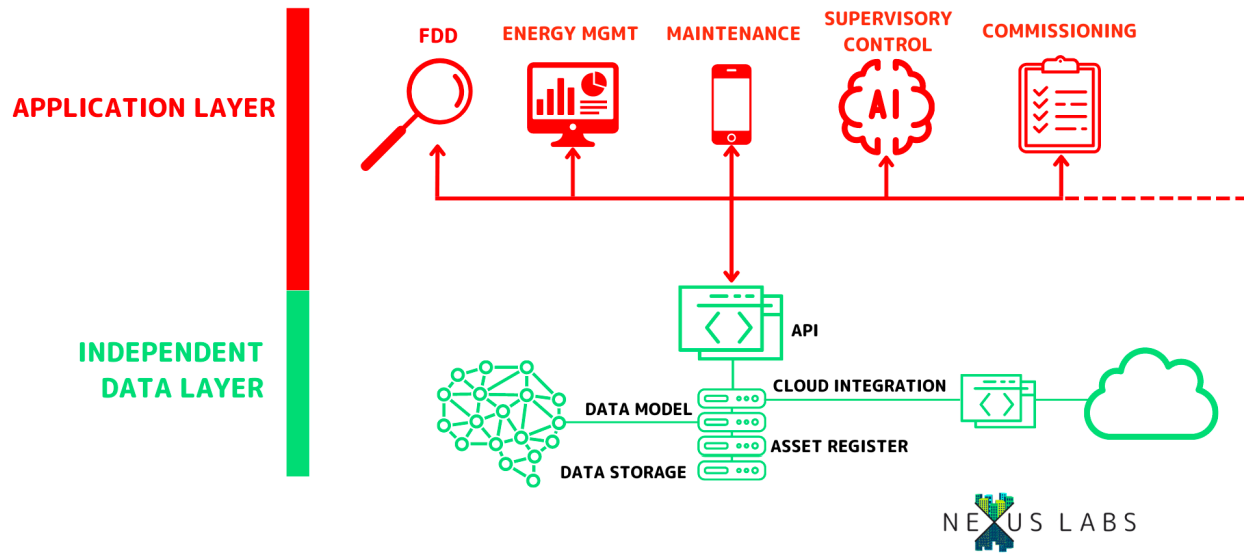
So what do we need that we don't already have?

First, most applications do just three things: they collect, visualize, and analyze data. And while the insights produced can be used to enhance a bunch of different workflows in the average portfolio, the user needs to exit out of the application to actually get sh*t done.

What if smart building applications were built around workflows instead of built around flooding us with "insights"? What if applications were focused on automating human tasks, helping reach outcomes easier, and enabling better collaboration?

Take fault detection and diagnostics (FDD), for example. Diagnostics can be used to enhance a bunch of different workflows in the average building, including energy management, capital planning, equipment maintenance, and commissioning. But FDD applications aren't built around any of those workflows, are they?

Shouldn't we have energy management, capital planning, equipment maintenance, and commissioning apps that are based on the actual workflow of each user persona and underpinned by the same FDD insights, context, and data set?



For example, if a facility manager receives a hot/cold call and opens it up in the maintenance app, shouldn't she see the active faults on upstream equipment, the status of preventative maintenance tasks, historical trends on how many people were in that space, the design occupancy of the space, and a calendar feed that shows how many people will be in the space for the rest of the week?

And when the FDD analytics discover an energy conservation measure she wants to implement, wouldn’t it be nice if she could send that into an energy management and/or capital planning

app, where her teammates can group it with similar measures into a project and analyze the financials, seek budget approval, seek financing, etc?

And wouldn't it be nice if the other apps were aware of the progress of implementation so they didn't bug me with new alerts about that opportunity? And when the energy conservation measure gets implemented, couldn't it trigger a workflow in the commissioning app, which would rely on the analytics in a slightly different way to track the completion, facilitate functional testing, and verify the results?

Okay, let's stop the daydreaming. In summary: we need persona-specific, contextually integrated applications that are designed around digitizing and automating human workflows.

Conclusion: The Messy Middle

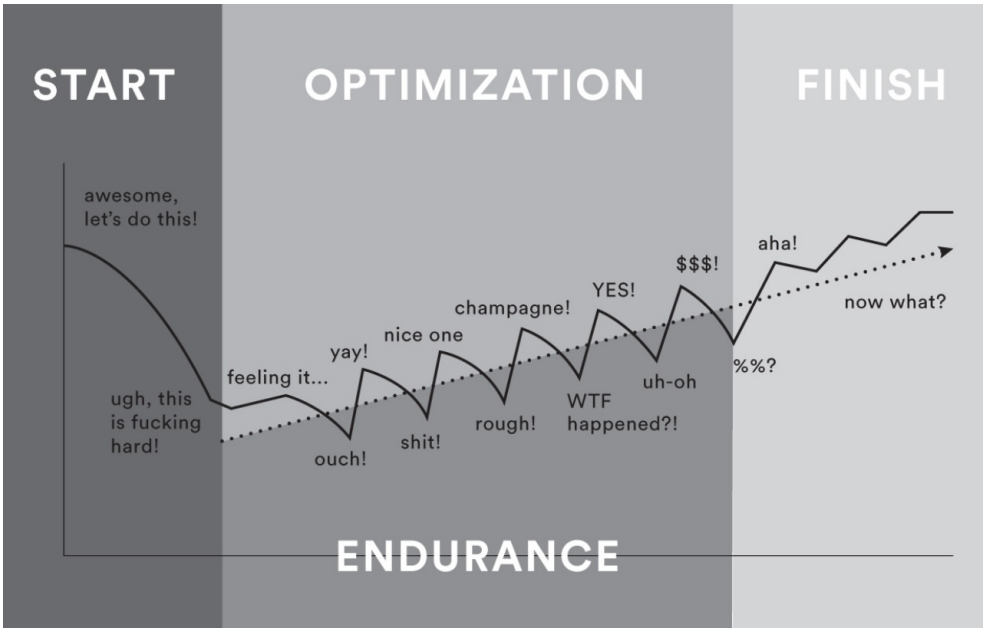
Let's conclude by collectively accepting one important truth: We live in a world where very few vendors or building owners are aligned with The Horizontal Architecture. Despite this vision for the future, the industry keeps on ticking. Old habits die hard.

If we look around, siloed point solutions are gaining market share each day. There are Device Layer vendors selling the mythical "Single Pane of Glass" application. There are "digital twins" and "building operating systems" and "smart building platforms" that combine many layers.

I'm not being pessimistic; I'm being realistic. Once we converge on a common vision, we then need to converge on a path to get there. And we definitely don't have that path yet. We lack a common plan for transformation.

This transformation will not happen overnight. To quote Nexus Pro member Greg Goelz, *"it is one step at a time and those waiting for the 'complete' solution will be waiting a VERY long time. Incrementally addressing key issues (by priority) while keeping sight of the complete solution is how transformation occurs."*

If the industry were a business, we'd be in what author and entrepreneur Scott Belsky has termed 'The Messy Middle'—that phase where the best strategy is to *"endure the downs (the incremental setbacks and struggles) and optimize the ups (everything and anything that seems to be working)".*



Source: TheMessyMiddle.com

So what are the incremental setbacks and struggles that are getting in our way?

#1: Marketplace confusion

There are many flavors of products that claim to provide some of the benefits and some of the capabilities we've outlined in this series. From application providers touting their API, to the infamous SPOG, to everyone claiming to be a platform. It's a murky situation out there for building owners.

#2: Fragmentation

Solving problems with smart buildings technology today involves too much switching. Between different tools; between different vendors and service providers. Tools aren't interoperable despite our many standardization efforts.

#3 Procurement

Building owners' processes for buying things must evolve with the evolution of technology. In the case of the transition to a horizontal architecture, they need to evolve to buying infrastructure layers instead of point solutions.

#4 The construction process

There's no better time than to build 'smart from the start', right? Unfortunately, our processes for building new buildings (and retrofitting old ones) don't produce smart buildings. Nexus Pro member Tom Balme produced a long list of challenges he's seen on construction projects recently:

- the time gap between specs and completion

- value engineering that removes key technologies
- technology providers working as 3rd or 4th tier subcontractors
- technology not completed/commissioned properly at completion
- the competitive race to the bottom among vendors and contractors

#5 The skills shortage

As I said at the end of 2021, workforce issues are everywhere. Consulting engineers need to upskill to specify the smart building. Building owner organizations need to upskill to manage the smart building. Everyone needs to train outsiders if we're going to grow the industry rather than pull from the same talent pool.

As we conclude Nexus Lore, we'd love to hear from you:

- What are the "ups" that seem to be working in your corner of the smart buildings industry?
- What can we do better as an industry to navigate this messy middle?

Let us know on LinkedIn.