



2026
DYNAMICS
CON
Vegas



2026 DYNAMICS CON



Sean Roach

**Unified Development Experience
(UDE) - One Year Later**

About Me

Connect with me on
LinkedIn



Sean Roach
Technical Consulting Manager



About Sean:

- Technical Solution Architect and Manager at New Arc Solutions
- 6 years in the D365 ecosystem. 9 full-cycle F&O implementations, 30+ clients across full-cycle and ongoing support
- Focus: Supply Chain, major distributors and manufacturers
- Particular interest in integrations and Azure Integration Services

About New Arc Solutions:

- Most of the team came from Blue Horseshoe Solutions - Architects of the WMS/TMS modules sold to Microsoft, the foundation of what is now Advanced Warehouse Management in F&O
- Focus on real client value. We get called in on the hard cases; rescue projects are a significant part of what we do



After Action Report

What We are Covering Today

Section	Slides	Time
Introduction	1-4	3-4 min
Section 1: What Are These	5-8	5-7 min
Section 2: Microsoft's Sales Pitch & Promises	9-12	8-10 min
Section 3: One Year Later (Good, Bad, and Ugly)	13-27	25-30 min
Section 4: Practical Recommendations	28-30	5-7 min
Q&A	31	10 min



What are These?

UDE

Unified Development Environment

VS

CHE

Cloud Hosted Environment



The Old Model (CHE's)

Where we came from:

- All-in-one Azure virtual machines deployed through Lifecycle Services (LCS)
- Everything self-contained on the VM: SQL Server, AOS, Visual Studio, X++ source code
- Client-hosted - required an active Azure subscription and ongoing infrastructure management
- Completely isolated from Dataverse - no connectivity between the dev environment and the Power Platform ecosystem
- Developers accessed it via RDP; VMs were frequently undersized by clients trying to manage costs



The New Model (UDE's)

What replaced it:

- Cloud-provisioned environment that pairs a stripped-down F&O instance with a live Dataverse instance
- Managed entirely through the Power Platform Admin Center (PPAC) - not LCS
- Architecture intentionally mirrors production topology - this unlocks capabilities we'll talk about
- Your local workstation is the dev machine: Visual Studio, debugging, and asset management all happen locally against the cloud environment
- No LCS. No Onebox. No client Azure subscription.



Side-by-Side: CHE vs. UDE

The architectural shift at-a-glance:

	Cloud Hosted Environment (CHE)	Unified Developer Environment (UDE)
Managed through	Lifecycle Services (LCS)	Power Platform Admin Center (PPAC)
Dev machine	Dedicated cloud VM (Azure)	Developer's local workstation
Dataverse	Isolated — no connection	Live paired instance
Azure subscription	Required (client-hosted)	Not required
Infrastructure management	Client / partner	Microsoft-managed
LCS dependency	Yes	No



MICROSOFT's Sales Pitch & Promises

4 Value Props

We will revisit in Section 3



Microsoft's Value Props

Eliminate the VM burden. Unify admin under PPAC

Value Prop 1: Eliminate the dedicated VM burden

- Microsoft's own docs describe the old model as "costly in addition to the purchase of licenses" and limited in Dataverse connectivity
- UDEs remove the Azure subscription dependency and the dedicated VM overhead, and the IT management that comes with it
- Developers work from their own local machines

Value Prop 2: Unify admin under PPAC, deprecate LCS

- LCS is being deprecated. UDEs are the first step toward a single lifecycle management surface across all D365 apps
- Framed as reducing friction and providing consistent operations across provisioning, copying, backup, and restore



Microsoft's Value Props

Close the topology gap. Simplify the cost model.

Value Prop 3: Close the dev-to-production topology gap

- For the first time, developer environments include a paired live Dataverse instance
- Enables real development and testing of Dataverse-dependent features: dual-write, Inventory Visibility, Planning Optimization, virtual entities
- First time developers can debug across F&O and Dataverse simultaneously in a dev environment

Value Prop 4: Simplify cost model via capacity-based-licensing

- Transition from environment-slot purchasing to capacity-based model consuming Dataverse and Operations database storage
- Framed as more flexible. No hard limit on environment count



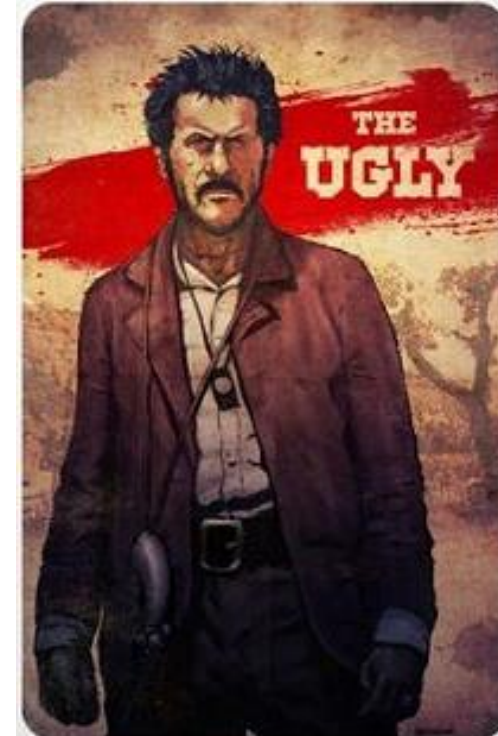
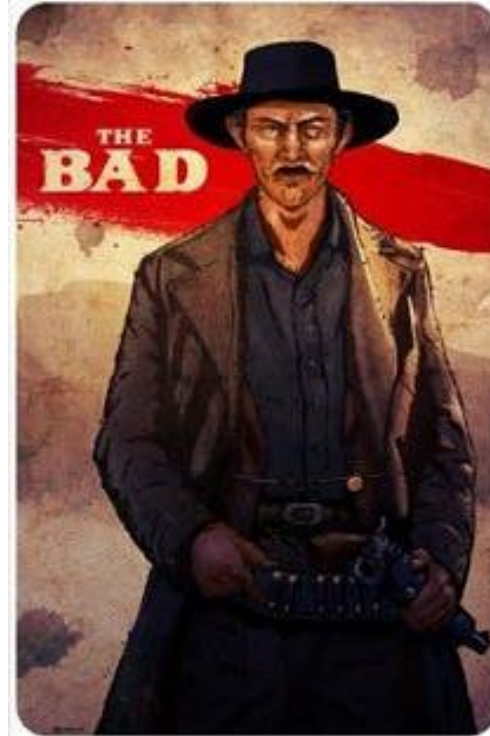
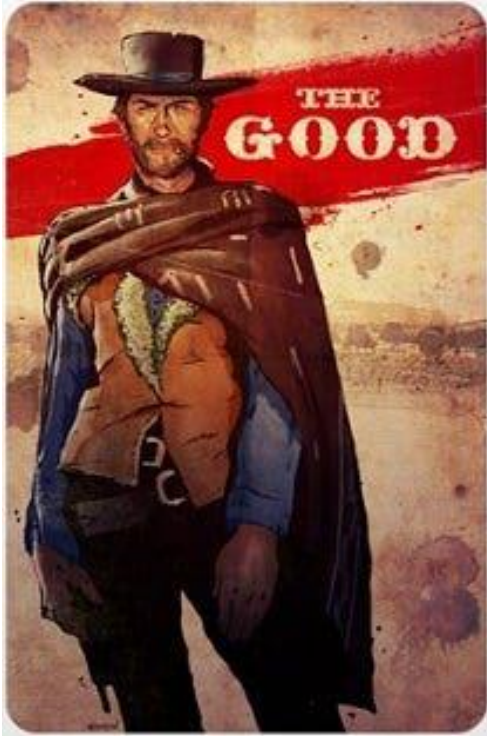
Setting the Stage

Four promises. Let's see what held up:

Value Prop	Promise	Reality
Eliminate VM burden	Simpler dev setup, lower overhead	Mixed – local compile has real tradeoffs
Unify under PPAC	Single surface, less friction	PPAC has reliability problems
Close topology gap	Dataverse parity in dev environments	This one delivered!
Simplify cost model	Flexible, capacity-based	Opaque, potentially more expensive



One Year Later – The Honest Review



One Year Later – The Honest Review

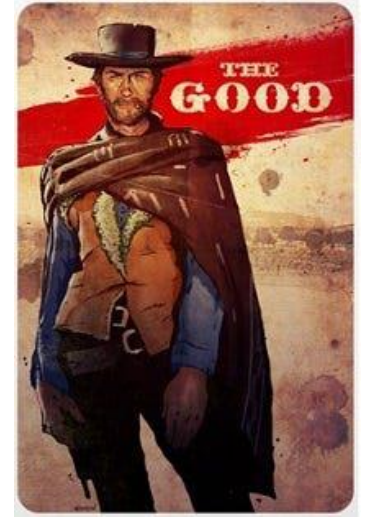


What actually works better

Debugging is Genuinely Better

This is the headline win

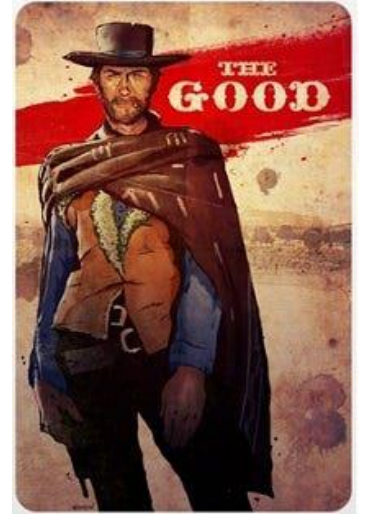
- Remote debugging is faster and more stable than the chronically undersized Dev VMs most clients were running (AOS crashes, out-of-memory errors, Visual Studio plugin slowdowns)
- Full access to Power Platform add-ins in a dev environment with minimal setup - this wasn't possible before
- **Dual-write debugging is a significant personal win:** you can see the exact order of operations F&O is executing on the dual-write side - well beyond what telemetry alone could show
- This level of visibility has never been available in a developer environment before
- If you work with dual-write or Dataverse-dependent features, this alone changes your workflow



Developer Workflow Quality of Life

The small wins that add up

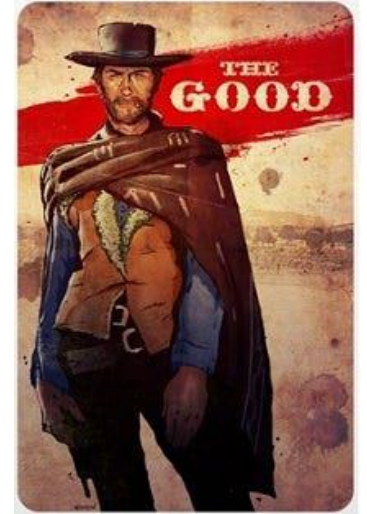
- Incremental deployments are fast and don't disrupt your standard workflow
- Project-level auto-deployment works well and reduces manual steps
- Multi-client, multi-version switching: flip of a checkbox vs. managing a suite of RDP sessions - easy to underestimate until you've lived the alternative
- Dramatically fewer Visual Studio crashes and "Finance and Operations plugin is slowing down Visual Studio" warnings
- "Works on my dev VM" surprises are reduced - architecture is closer to production topology



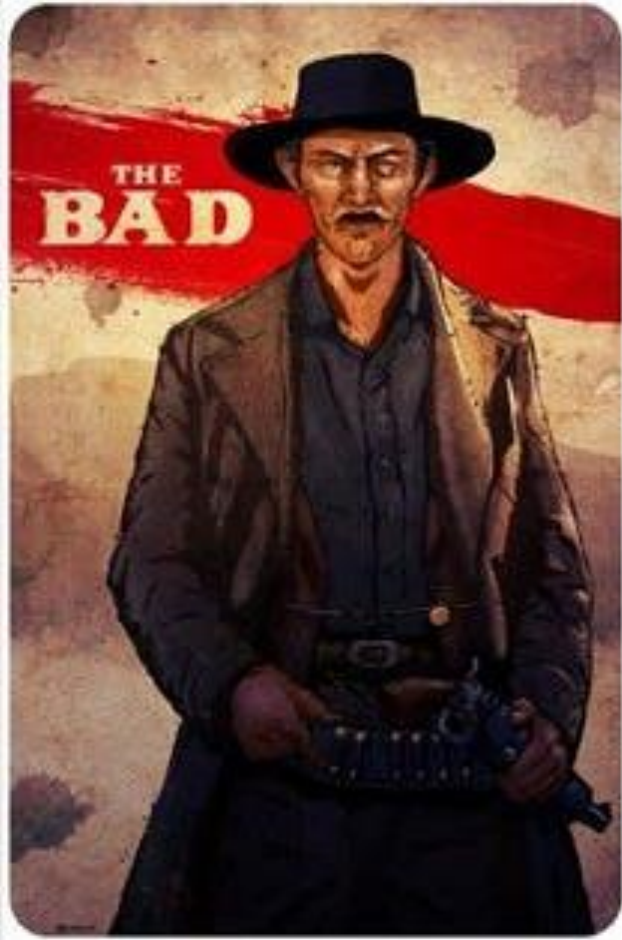
Environment Management Wins

Copy, clone, and manage with less pain

- One-click copy of both Dataverse and F&O together — this is a meaningful new capability
- Transactionless copying is net-new: useful for reducing sandbox storage consumption and improving environment performance by copying data without historical transactions
- Production-mirrored topology means what you build in dev is more likely to behave the same in sandbox and production



One Year Later – The Honest Review

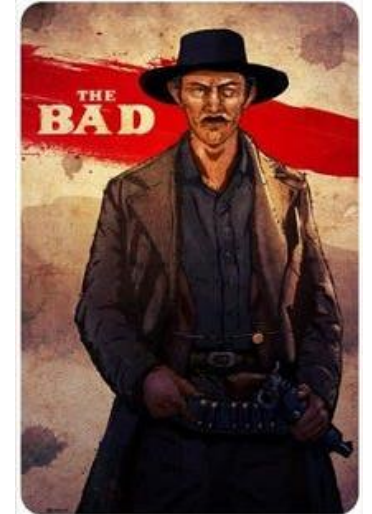


Where the real gaps are

Deployable Packages – A Critical Gap

One-time runnable scripts have no clean home

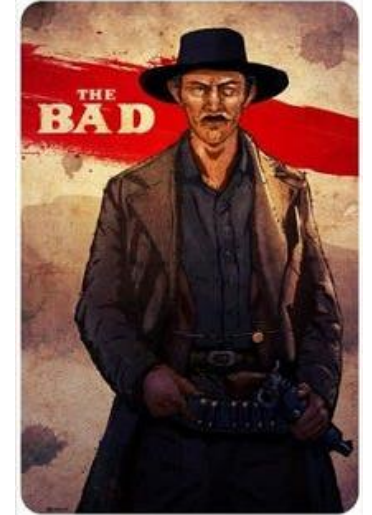
- Deployable packages from Visual Studio are now incremental only. There is no UI distinction to indicate this
- There is no mechanism to create packages for one-time runnable scripts
- This is not a minor inconvenience: one-time scripts are a standard tool for post-go-live critical fixes - data corrections, configuration patches, emergency remediation
- No clean workaround currently exists
- This is an artificial limitation - all the build components needed to produce a full package exist on the local workstation; Microsoft has simply not exposed a path to build one
- A CHE doesn't have this problem



Storage and Update Management

30-45GB. Right now. Before you can work.

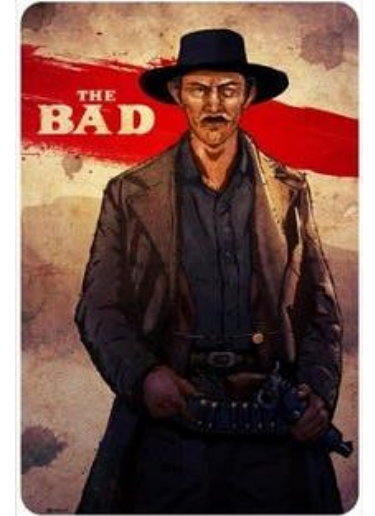
- Every metadata version requires a full separate download of the metadata package, reference database, and VS extension - approximately 30–45GB
- A minor quality update is treated identically to a full application update
- Updates push approximately once or twice a month and are not optional
- How it surfaces in practice: A developer opens Visual Studio and is prompted to download metadata...That's a 30–45GB download. Right now. Before they can do anything.
- An in-place update requires approximately 60GB of free disk space simultaneously (existing + new); you can delete the old version first, but that's a manual step
- No proactive communication when updates push: no email, no PPAC notification - you find out when the prompt appears
- Compounds quickly in a multi-client shop maintaining multiple metadata versions simultaneously



Documentation Hasn't Kept Up

The environment ID story – tell this as a story

- Following Microsoft's own documentation to install Inventory Visibility in a UDE, you arrive at the step that says to retrieve the F&O environment ID from LCS
- LCS doesn't exist in this world — and there is no equivalent surface in PPAC
- After approximately an hour of digging: an undocumented internal Dataverse API that returns the integration details:
- `{envurl}/api/data/v9.2/RetrieveFinanceAndOperationsIntegrationDetails`
- Nothing in the documentation points you there. First-party add-in docs still assume LCS exists.
- This is a symptom of a broader pattern: the product moved forward, the documentation ecosystem didn't fully follow it
- When you hit a wall, your support path (as we'll cover) is also not reliable

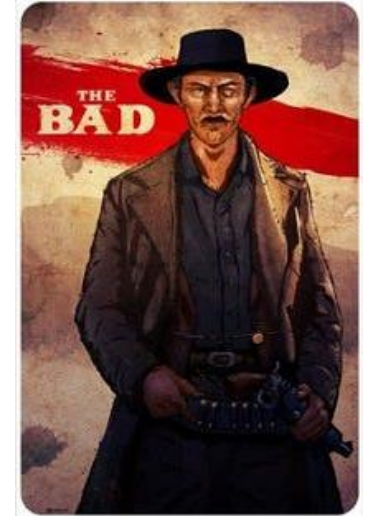


Compilation Tradeoffs & Debugging Limits

The fine print on local development

Compilation on the local workstation:

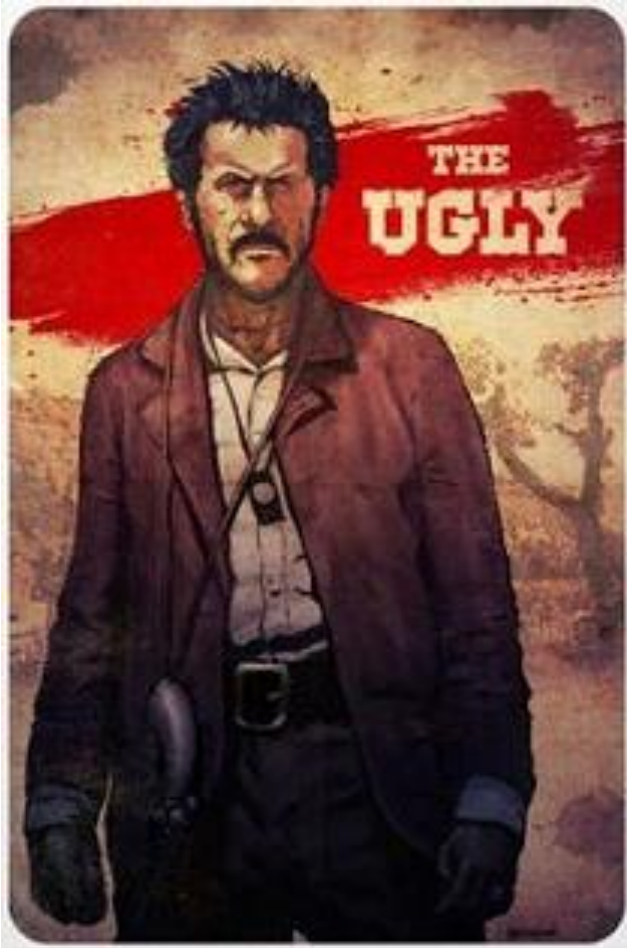
- Neutral-to-good for developers on powerful, dedicated machines
- Meaningful performance issue for large codebases
- Local workstations carry OS drift, other installed software, and inconsistencies. Introduces a class of build failures that a dedicated VM simply doesn't have



Debugging limitations that didn't exist with CHEs:

- Detaching from the debugger frequently kills the AOS — disruptive mid-session
- Batch debugging is not supported — significant gap for complex batch job development and troubleshooting
- C# class libraries do not have symbols included when deployed — these cannot be debugged
- You can no longer debug against a different sandbox since no web.config

One Year Later – The Honest Review



Where trust breaks down



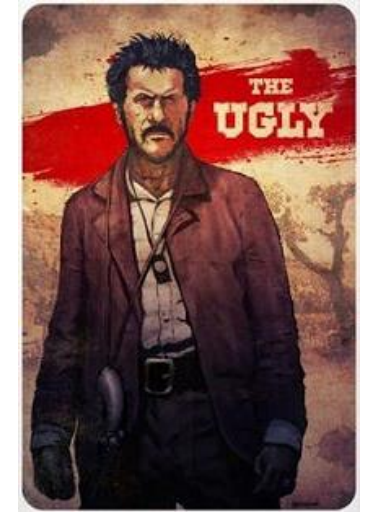
Cost is a black box

The math exists. Finding it requires a YouTube video

Problem 1: Discovery: The cost is not in PPAC. Not in Partner Center. Not a clear line item on the invoice. This is not a permissions problem - we went looking specifically to answer this question for this talk. The authoritative source for how UDE pricing actually works is a FastTrack TechTalk on YouTube. Not official documentation.

Problem 2: The Math: Pricing is capacity-based. Dataverse capacity is \$42/GB. A typical UDE for our clients runs 7–10GB of Dataverse usage - roughly \$300–\$420/month per environment. A Cloud Hosted Environment on an appropriately sized Azure VM with auto-shutdown comes in cheaper, with a cost you can predict upfront.

Problem 3: Variability: Unlike a VM where cost is essentially fixed at provisioning, Dataverse capacity grows with data over time. The monthly cost is not static. Monitoring requires checking each individual environment in PPAC manually — there is no aggregated billing dashboard.



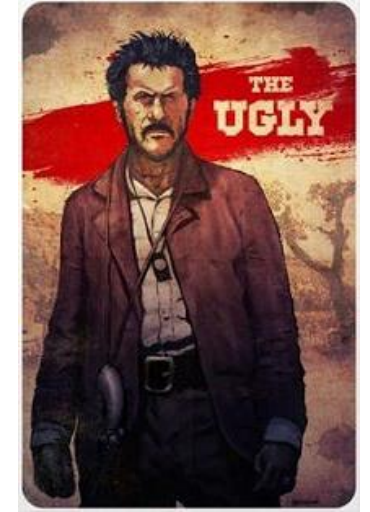
Cost is a black box

The math exists. Finding it requires a YouTube video

Problem 1: Discovery: The cost is not in PPAC. Not in Partner Center. Not a clear line item on the invoice. This is not a permissions problem - we went looking specifically to answer this question for this talk. The authoritative source for how UDE pricing actually works is a FastTrack TechTalk on YouTube. Not official documentation.

Problem 2: The Math: Pricing is capacity-based. Dataverse capacity is \$42/GB. A typical UDE for our clients runs 7–10GB of Dataverse usage - roughly \$300–\$420/month per environment. A Cloud Hosted Environment on an appropriately sized Azure VM with auto-shutdown comes in cheaper, with a cost you can predict upfront.

Problem 3: Variability: Unlike a VM where cost is essentially fixed at provisioning, Dataverse capacity grows with data over time. The monthly cost is not static. Monitoring requires checking each individual environment in PPAC manually — there is no aggregated billing dashboard.

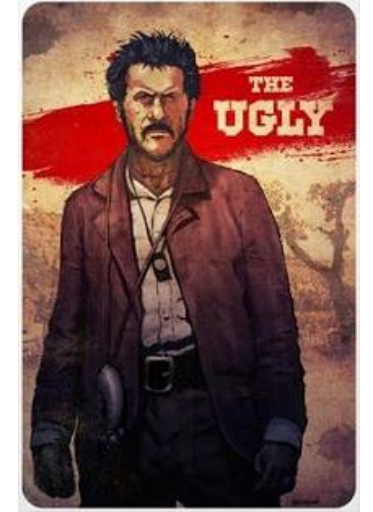


The Cost Conversation You Can't Have

You can't give a client a good number before their first UDE

The estimating gap:

- With a CHE: open the Azure pricing calculator, enter the VM SKU, region, and expected uptime → hand a client a defensible number before you spin up a single environment
- With a UDE: Dataverse capacity at go-live is unknown for a new client. You can offer a range. That's it.



The hidden license cost most clients won't be expecting:

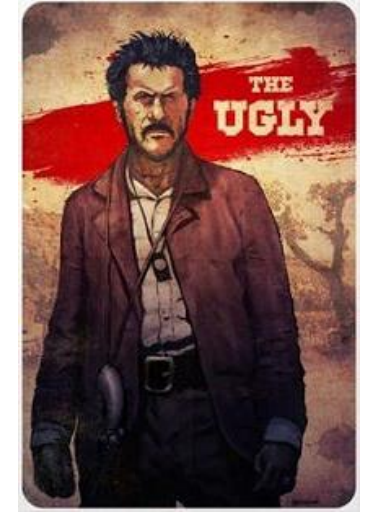
- PPAC does not support guest accounts — every developer working in a UDE must have a licensed account in the client's own tenant
- With CHEs: a developer carried a single license with their employer and used it across every client engagement
- With UDEs: each client is now effectively paying for that developer's license for the duration of the engagement — on top of the capacity cost
- This conversation has to happen before the first environment is spun up

Support Isn't Ready & PPAC Reliability

Blocking issues have an unreliable support path

Support:

- Tier 1 engineers frequently don't know UDEs exist — reaching someone who understands the request requires multiple re-routings
- Tier 1 initial responses are generated by Copilot, which has no knowledge of UDEs — the first response you receive is often confidently wrong
- If you hit a blocking issue in a UDE, your support path is genuinely unreliable
- This matters more than it would with a CHE — you don't have access to the underlying infrastructure to diagnose or work around problems yourself



PPAC reliability:

- Credential caching in Visual Studio behaves erratically beyond normal VS quirks
- PPAC does not support guest accounts — developers must operate under a client-provisioned account
- PPAC copy screens display inaccurate information about what is and isn't being copied
- When you do copy environments, sometimes add-ins are cloned as well but do not show up as installed in PPAC

PRACTICAL RECOMMENDATIONS

So What Do We Actually Do?

Is my recommendation to avoid these?



Recommendations: Developer Experience First

Setting cost aside: UDEs win — with one important carve-out

- Hard to make a strong case for CHEs as a primary developer environment anymore
- Constant uptime, ease of deployment, local dev experience, and full Power Platform add-in access make UDEs the better daily driver in almost every scenario

Keep at least one CHE in reserve for:

- One-time runnable script development (until Microsoft closes this gap)
- Hotfix and emergency debugging against a Tier 2 sandbox

Recommended posture: UDEs as primary. One CHE kept off, in reserve, spun up only when needed.



Recommendations: Bringing Cost Back In

It is genuinely client by client

- CHEs remain a known quantity — transparent cost structure, mature tooling, well-understood workflows across the industry.

No compelling reason to migrate if a client doesn't have the workflows that benefit from UDE advantages.

Factors supporting no rush:

- LCS deprecation deadlines keep getting pushed — no risk of being caught flat-footed on CHEs
- Feature parity between PPAC and LCS is still incomplete, with no clear roadmap to address it
- The next 12 months will not put anyone meaningfully behind

Scenarios where UDEs make strong sense right now:

- Active dual-write, Inventory Visibility, Planning Optimization, or other Dataverse-dependent development
- Clients wanting to reduce or eliminate Azure infrastructure management burden
- Net-new implementations with no existing CHE investment to protect



Q&A



Sean Roach

Technical Consulting Manager



www.newarc.solutions



Connect with me on

LinkedIn

