



Deploying your technology

... *smartly*

inx

inx limited

www.inx-systems.com / www.inxware.io

Deploying your Technologies

Integrate to win

“~ 75% of embedded software development time is typically spent on integration.”

- Thwarts technology adoption
- Slows product development
- Increases time to production revenue
- Risks failure to launch products
- Reduces product quality
- Dilutes features ultimately deployed
- Subtracts from product and system design
- Maintenance re-integrations repeats QA

Inxware makes your technology more:

- **Accessible** : no-code applications
 - **Integrated** : 100s of supporting components
 - **Configurable** : parameterised & re-codable.
 - **Deployable** : Seamless PoC to production path.
 - **Maintainable** : Built on Universal Open Source
-

inxware Enabled Devices



Industrial



Consumer



Edge Compute



Smart Buildings

- **ARM-A/M-series:** FreeRTOS, Zephyr, Linux, Android & Bare-metal
- **x86/AMD:** Linux, Windows, Unity
- **ESP32:** FreeRTOS/IDF

inxware has supported business critical IoT services and consumer products for over a decade with multiplatform products from 112 core compute DSPs to \$2 MCUs.



Eval Boards, SOMs, SBCs



Accelerating Technology Deployment

Integration first

- **Engineered Components**
 - consistent component APIs
 - synchronous & asynchronous
 - open & closed source C/C++
- **Application Dev Kit (ADK™)**
 - model-based implementation
 - System-level **debugging**
 - **UI** Layout add-ons
 - **AI** model deployment
- **Multi-platform**
 - concurrent HW/SW engineering
 - **future-proof** hardware options
 - **multi-OS** and board SDKs
 - multi-target **CI** build system.

Building Applications with Lucid ADK™

Component Library:
100s of built-in functional components and software constructs
Extensible in C or graphical sub-assemblies

3D Graphical Workspace:
Your application and business logic defined & documented as you go.

The screenshot displays the Lucid ADK software interface. On the left, a 'Component Library' pane shows various functional blocks like 'Latching AND', 'TextSwitch', 'Switch', 'Counters/Iterators', and 'Boolean Control'. The central workspace shows a state machine diagram with two states, 'State A' and 'State B', connected by transitions. A 'Timer' component is used to trigger transitions. On the right, a 'Debugger' window shows the 'time_clock' component's properties, including 'Block Type Properties', 'Instance Info', and 'Parameters'. The 'Parameters' table is as follows:

name	type	default	min.	max.	description
Time Interval (uSecs)	int	1000000	1	2147483647	Time interval between events.

Debugger:
Connect to a device running eRT and see every event and data value in real-time or stepped modes

Component Documentation

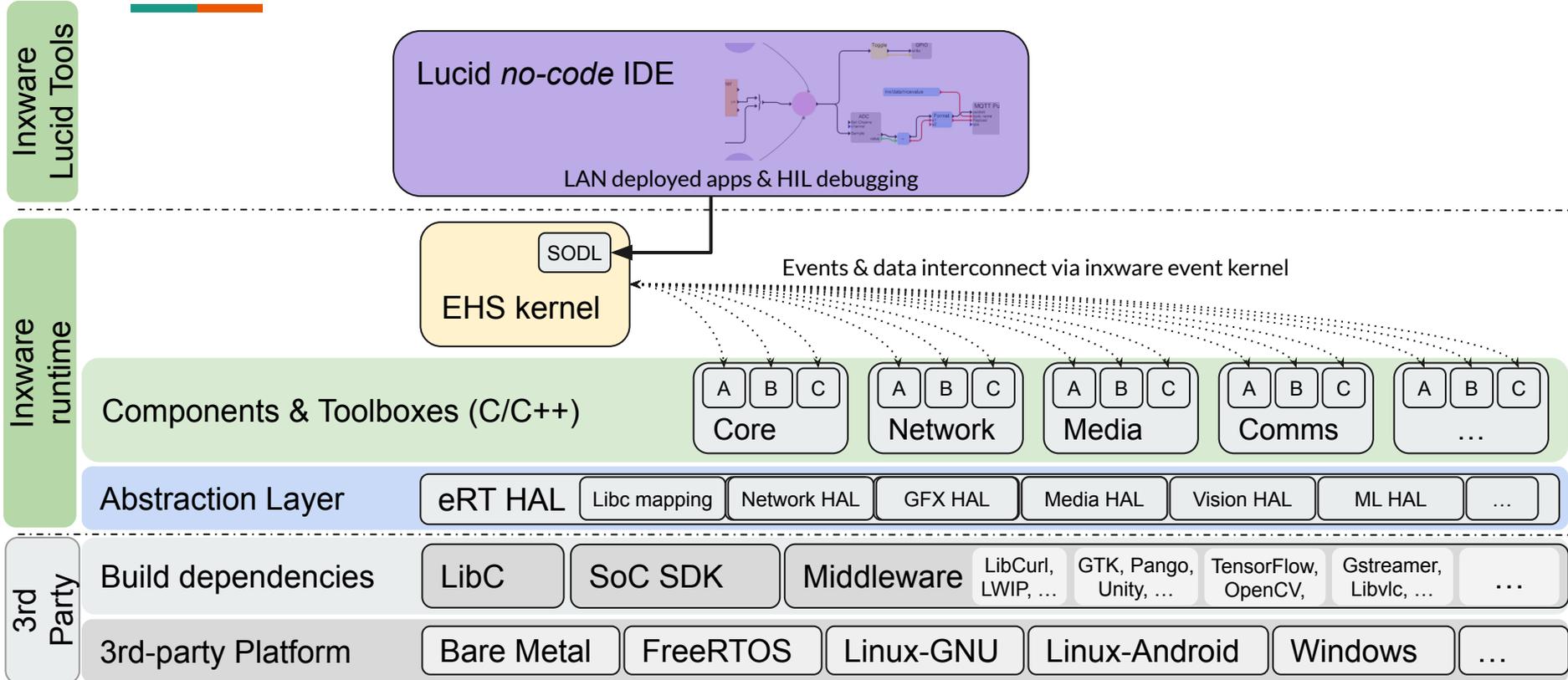
Every built-in component is provided with HTML manuals and test examples.

Separating Concerns

Don't mix firmware with
application development!

- **ADK IDE - for applications**
 - Real-time embedded applications
 - Data-rich & connected products
 - Hardware-in-the-loop debugging
 - OTA app deployment
 - Platform independent executables
- **ADK™ runtime firmware**
 - Generalised **event driven** runtime
 - Unique **abstraction** of executable code
 - **Lightweight** embedded
 - **Secure**, realtime & **robust** platform
 - Field proven for **15 year operation**
 - Portable to **multi-hardware** & OS.
 - **Extensible** component library

Inxware's Open Source Runtime Stack



“Inxware solves your systems integration problems and provides your customers with the tools they need to ship your technologies.”

Thanks for Viewing!

www.inxware.io - products

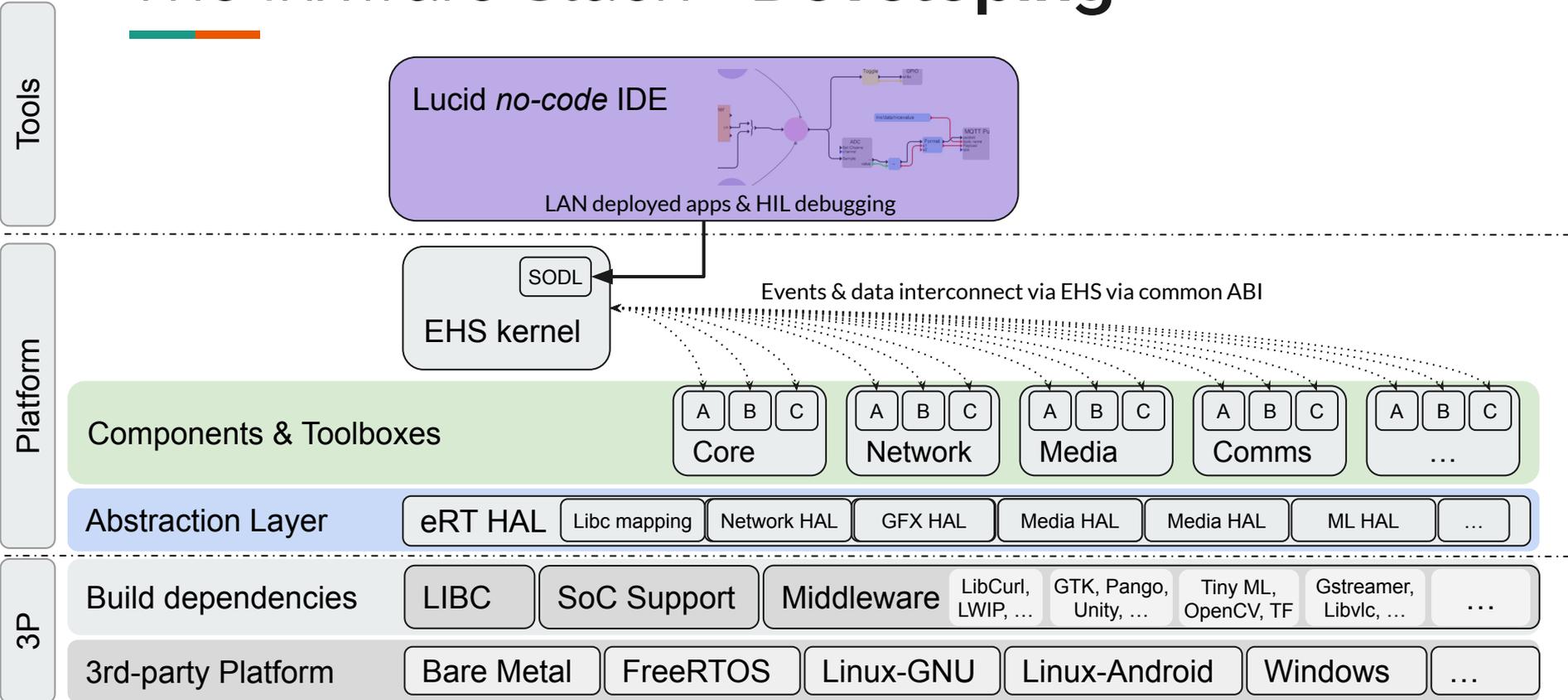
www.inx-systems.com - services



Inxware's extendible architecture

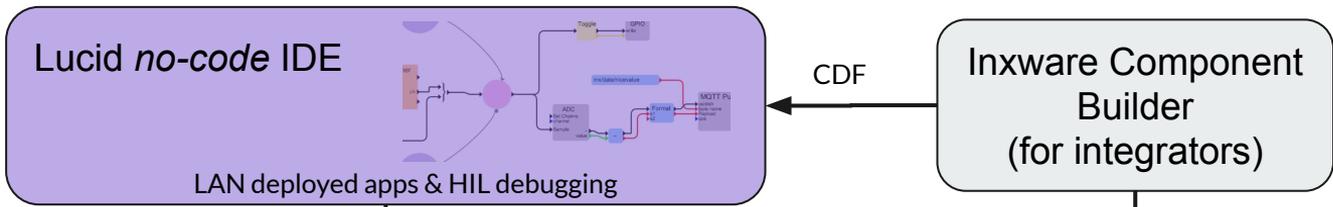


The Inxware Stack - Developing

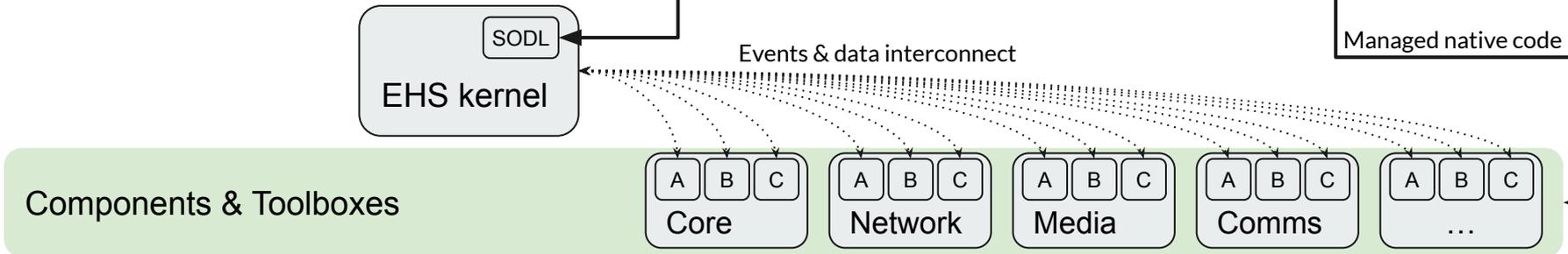


The Inxware Stack - Extending

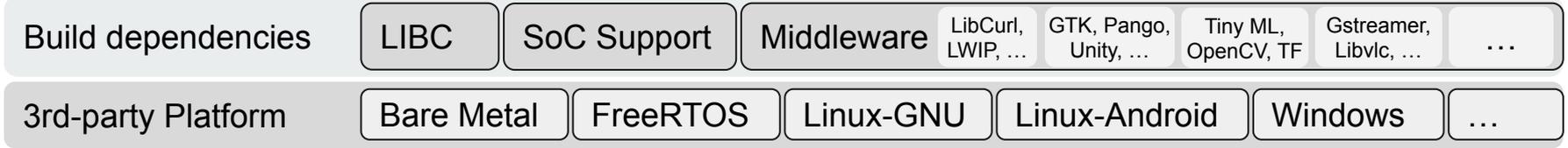
Tools



Platform



3P



Thanks for Viewing!

www.inxware.io

www.inx-systems.com

