



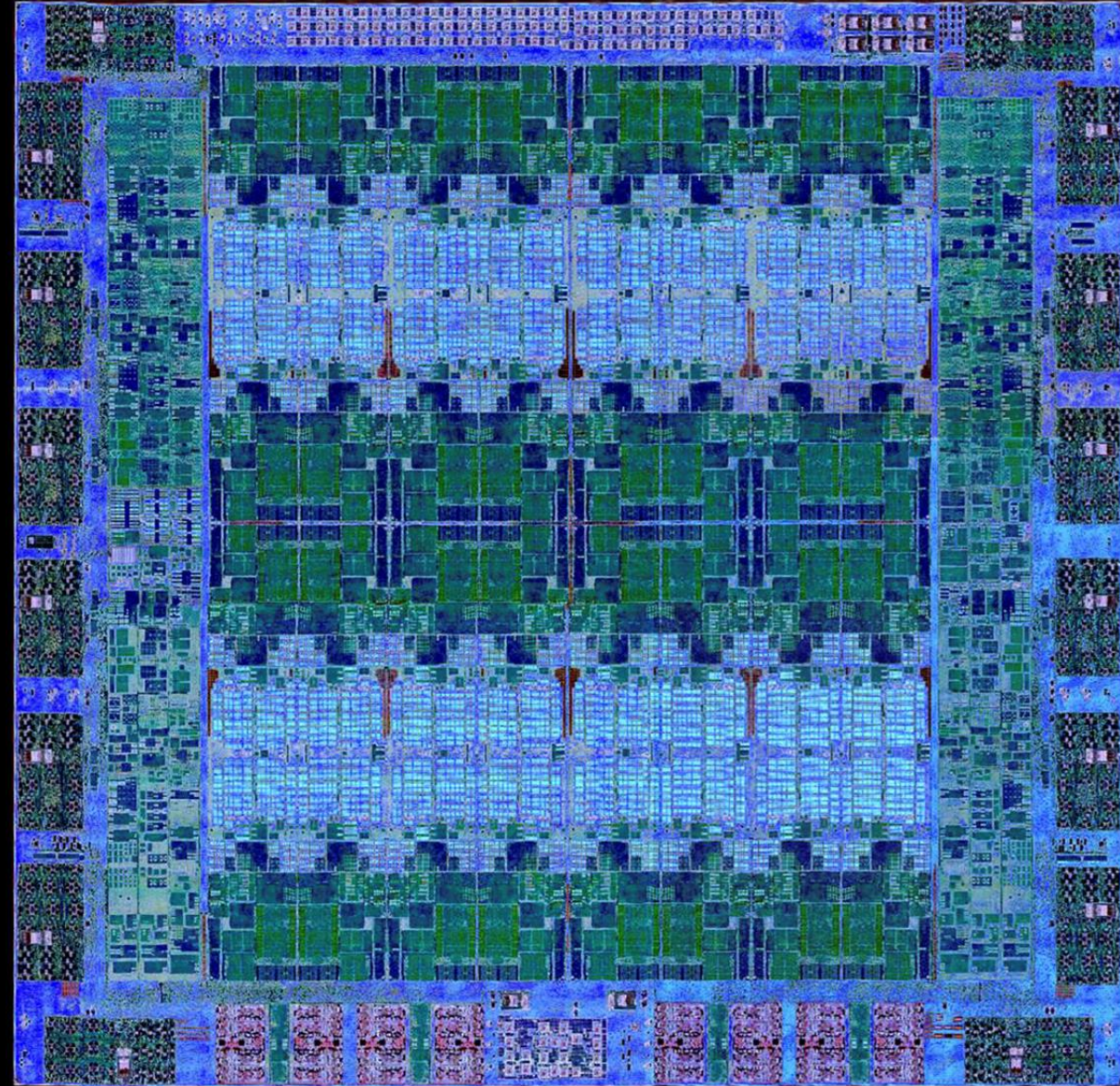
Coming back home

# Sean Flanagan

Global Vice President for Technical Sales  
IBM Cloud & IBM Power

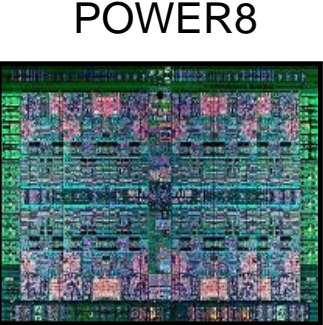


# Power Processor Roadmap

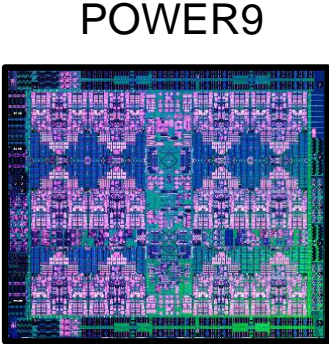


# IBM Power Processor Roadmap

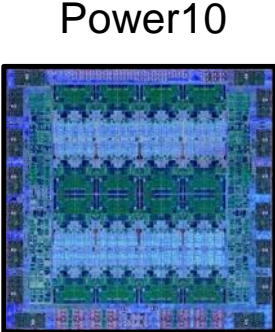
## Continuous Platform Innovation and Leadership



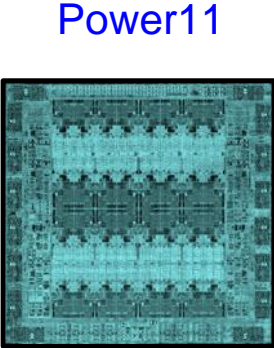
Powerful SMT8 Core  
Enterprise Scaling  
Big Data Optimized  
Agnostic Memory



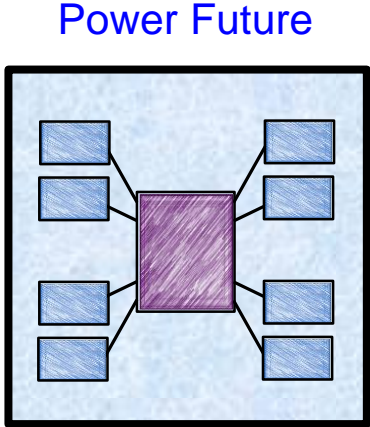
Modular Core Design  
Accelerator Attach  
(NVlink, OpenCAPI)  
Data Plane Bandwidth  
DDR & CDIMM Memory



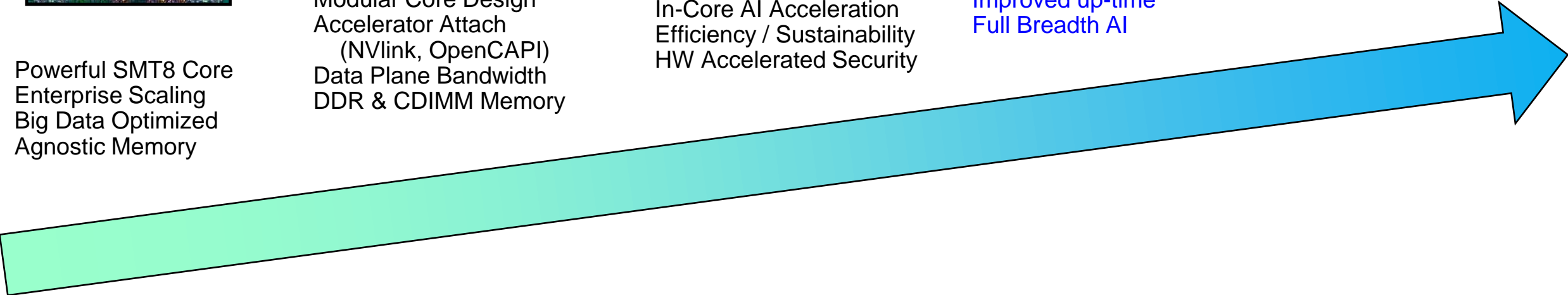
Core / Thread Strength  
Socket Performance  
In-Core AI Acceleration  
Efficiency / Sustainability  
HW Accelerated Security



Core / Thread Strength  
Socket Performance  
Enterprise Scaling  
Energy Optimization  
Improved up-time  
Full Breadth AI



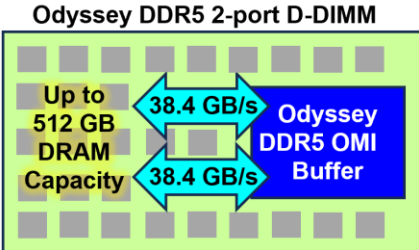
(Under development)



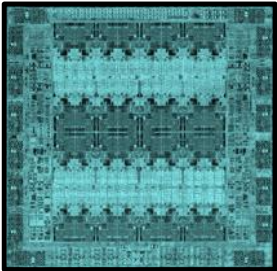
# Power11: Full Stack Innovation and Cross-Optimization

**Memory Architecture**  
**Energy / Thermal Infrastructure**

**Processor Architecture**  
**Socket-level Packaging**  
**Semiconductor Technology**



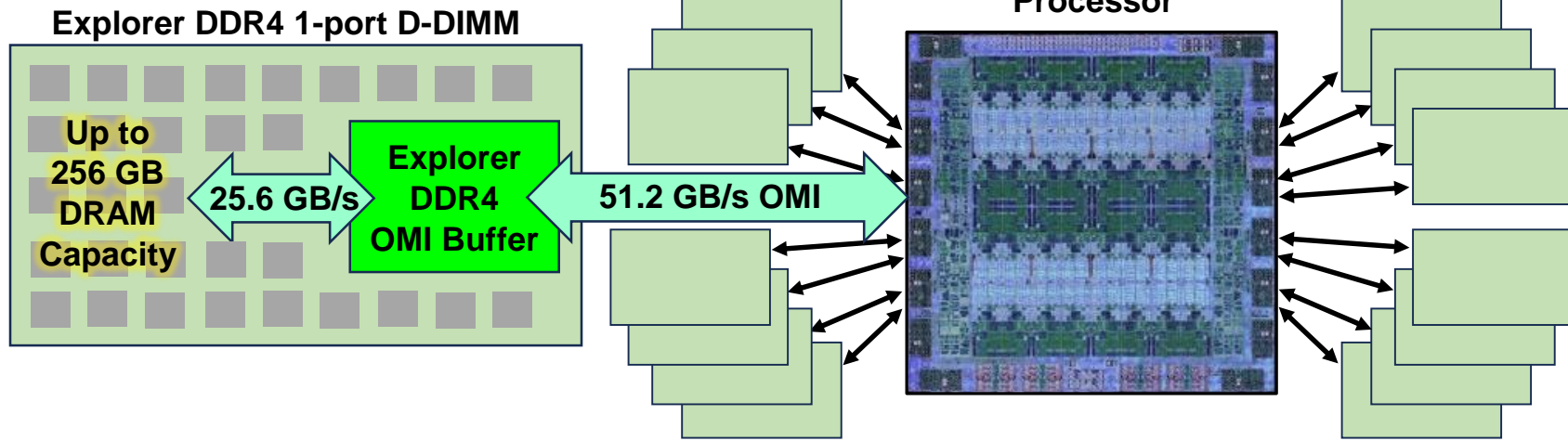
**Agnostic, 3x Pipes, 2x Capacity**  
**Advanced Cooling Technology**



**Improved Thread, Core, Capacity**  
**2.5D Stacking: Energy Optimization**  
**Samsung Foundries Enhanced 7nm**

# Memory Technology: OMI DDR5 Performance, Scaling, Economics

## 2021

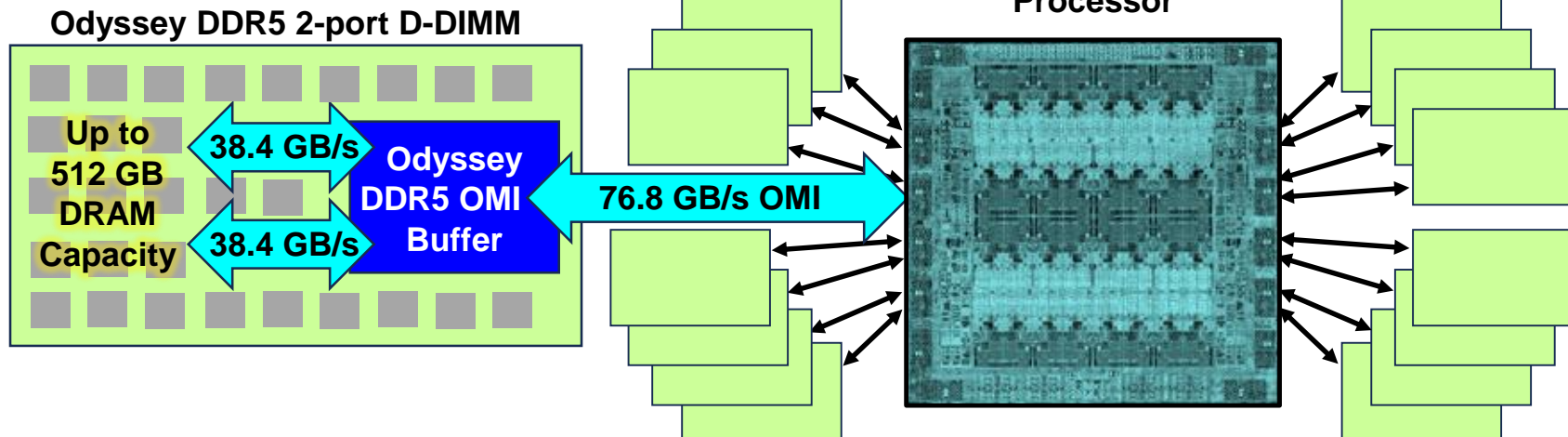


400 GB/s DRAM Bandwidth / Socket

4 Terabyte DRAM Capacity / Socket

700-800 GB/s System Coherence Flow

## 2025



1200 GB/s DRAM Bandwidth / Socket **3x!!!**

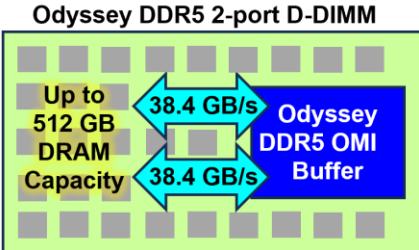
8 Terabyte DRAM Capacity / Socket **2x!!**

1000 GB/s System Coherence Flow **1.3x!**

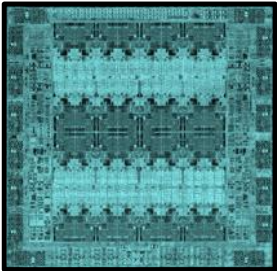
# Power11: Full Stack Innovation and Cross-Optimization

**Memory Architecture**  
**Energy / Thermal Infrastructure**

**Processor Architecture**  
**Socket-level Packaging**  
**Semiconductor Technology**



**Agnostic, 3x Pipes, 2x Capacity**  
**Advanced Cooling Technology**



**Improved Thread, Core, Capacity**  
**2.5D Stacking: Energy Optimization**  
**Samsung Foundries Enhanced 7nm**

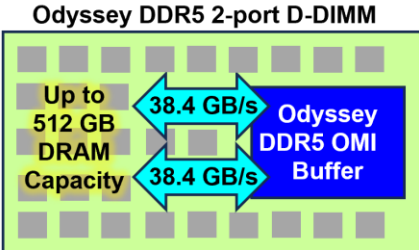
# Power11: Full Stack Innovation and Cross-Optimization

**Platform Capabilities**

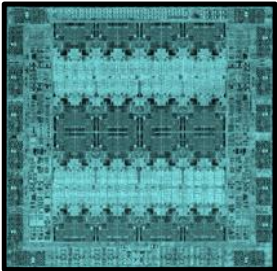
**Memory Architecture  
Energy / Thermal Infrastructure**

**Processor Architecture  
Socket-level Packaging  
Semiconductor Technology**

**Uptime, Energy Mgmt, Quantum-safe security**



**Agnostic, 3x Pipes, 2x Capacity  
Advanced Cooling Technology**



**Improved Thread, Core, Capacity  
2.5D Stacking: Energy Optimization  
Samsung Foundries Enhanced 7nm**



# Power11: Full Stack Innovation and Cross-Optimization

**AI Acceleration**

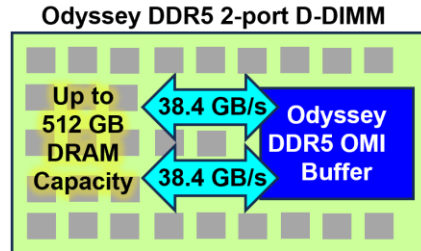


**Accelerated AI (all systems)  
IBM Spyre Accelerator  
Optimized for Inference**

**Platform Capabilities**

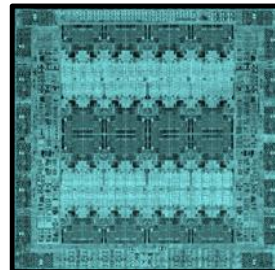
**Uptime, Energy Mgmt, Quantum-safe security**

**Memory Architecture  
Energy / Thermal Infrastructure**



**Agnostic, 3x Pipes, 2x Capacity  
Advanced Cooling Technology**

**Processor Architecture  
Socket-level Packaging  
Semiconductor Technology**

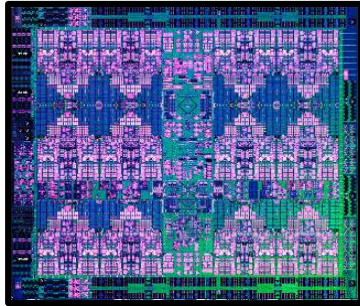


**Improved Thread, Core, Capacity  
2.5D Stacking: Energy Optimization  
Samsung Foundries Enhanced 7nm**

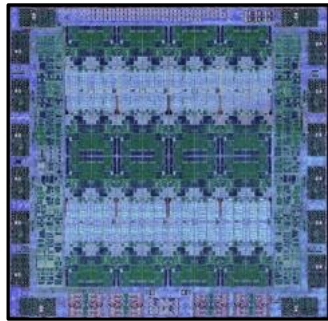
AI Improvements: Accelerated support for Large Models and Model Tuning



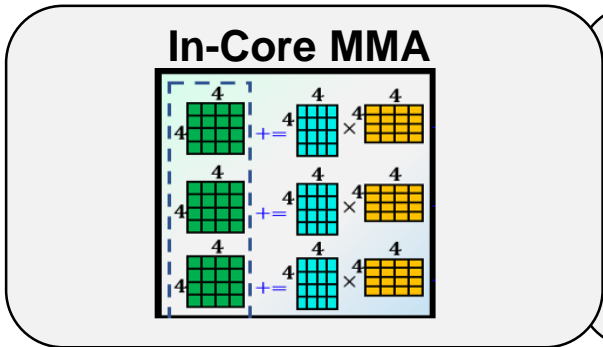
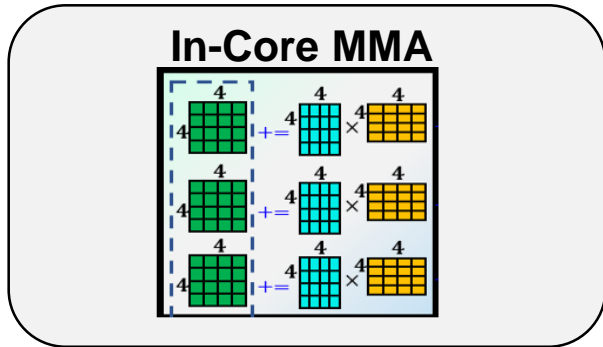
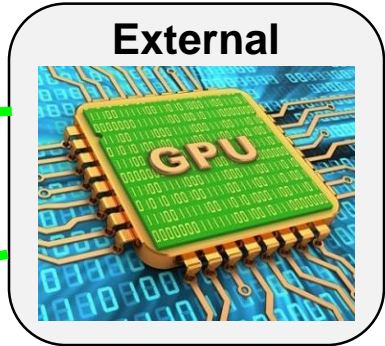
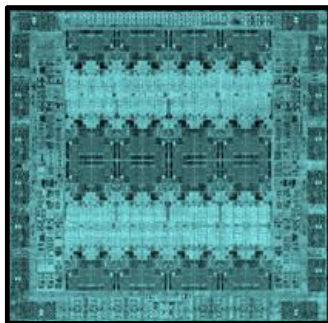
POWER9



Power10



Power11

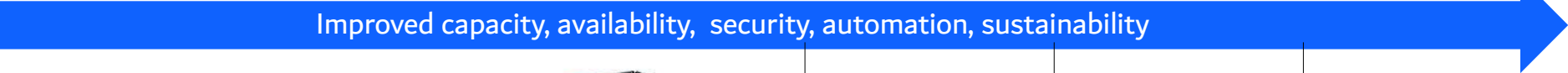


# IBM Power Platform: 10 Year Innovation Outlook

Co-creating with customers, in partnership with IBM R&D, to remain the most trusted and open computing platform for mission critical scalable transaction processing and data serving workloads

2017

2030+



IBM Power8

Big Data Optimized  
Live movement of HANA instances between servers  
AIX Oracle 16 TB RAM per LPAR and Live Kernel Update  
OpenPOWER



IBM Power9

Off-chip Accelerated AI  
World's 2 Fastest Supercomputers  
Fully shared resources for HANA & up to 16 virtualized HANA production DBs  
AIX 32 TB RAM per LPAR



IBM Power10

Accelerated AI with AI-optimized ISA  
60% less energy relative to Power8  
Memory Encryption and Quantum Safe Cryptography  
Scaling to 40TB SAP HANA LPAR  
AIX Live Update of Kernel Tunables



IBM Power Next

Zero planned downtime

Automated Error Resolution

Delivering on ESG Requirements

Quantum Safe Platform

Best in class HANA TCO



IBM Power Next +1

MultiKey Mem encryption

Accelerated and Simplified AI

Quality of Service through full stack isolation

Disaggregated Memory for TCA

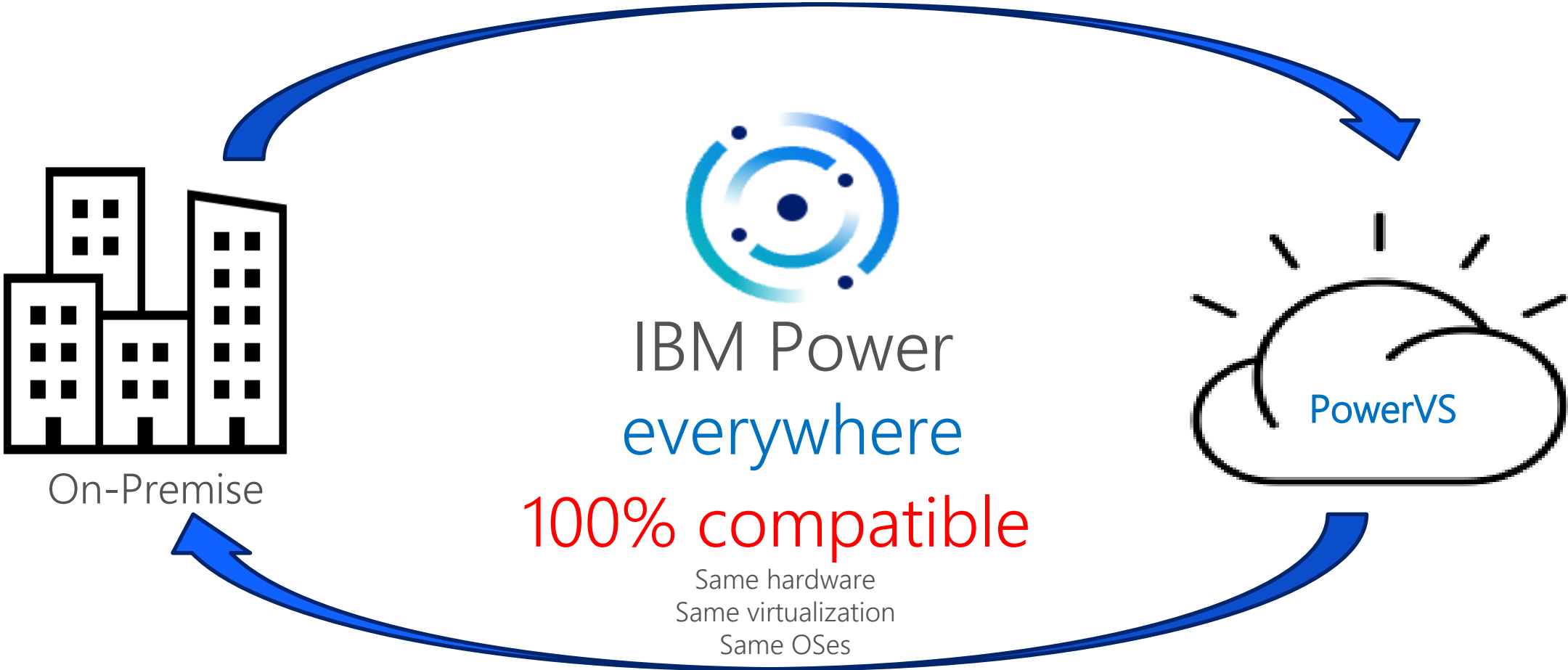
Quantum Integration



IBM Power Next +2

Transparent tiered memory support

# Hybrid is the new future



**IBM i** **AIX** **Linux**

Do we Have / Want it locally

**IBM i** **AIX** **Linux**

AIX, IBM i & Linux on Power  
Are available in the cloud

## Highlights

# 21

Data centers across the globe and growing

# 650+

Customers deployed Production, HA/DR, and Dev/Test use cases

### Key Workloads

- Certified SAP IaaS
- SAP NetWeaver and S4/HANA
- Oracle Supported
- IBM i workloads
- Epic - Healthcare (training case)
- Red Hat OpenShift
- IBM Cloud Paks
- OS: AIX, IBM i, Linux

### Comprehensive Compliance

- GDPR
- SOC 1 Type I and II
- PCI DSS Certification
- ISO 27K

### Why It Matters

- [Frictionless expansion and migration between on-premises and Power Virtual Server.](#) Identical architecture with enterprise Power Systems on-premises from microprocessors, firmware, PowerVM, PowerVC, dual VIOS to SAN storages.
- [Same workloads supported on-premises are supported in Power Virtual Server e.g. Oracle, SAP, IBM i, RedHat OpenShift](#)
- Secure Infrastructure as a Service, Resilient platform for mission critical workloads, Flexibility to scale on-demand, Cloud consumption model to pay for use, reduce CapEx, specialized skills not required, Cloud Native development on Power

# IBM Power Virtual Server

Worldwide deployment locations

21 data centers in 9 countries, across 10 regions and continuing to expand!

● Worldwide deployment locations

## Americas (4 Regions)

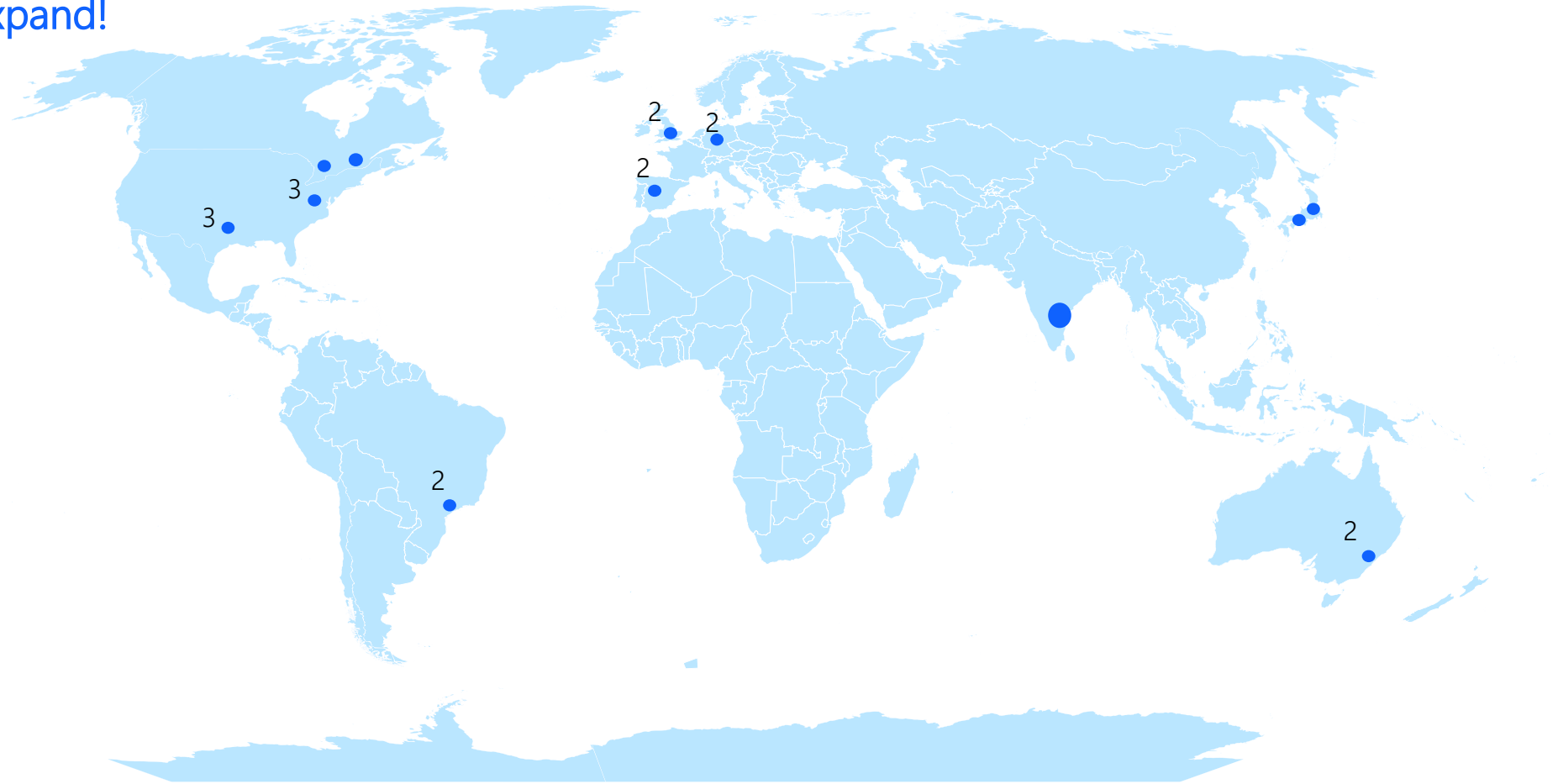
- Dallas (3)
- Washington DC (3)
- Toronto
- Montreal
- Sao Paulo (2)

## EMEA (3 Regions)

- Frankfurt (2)
- London (2)
- Madrid (2)

## APAC (4 Regions)

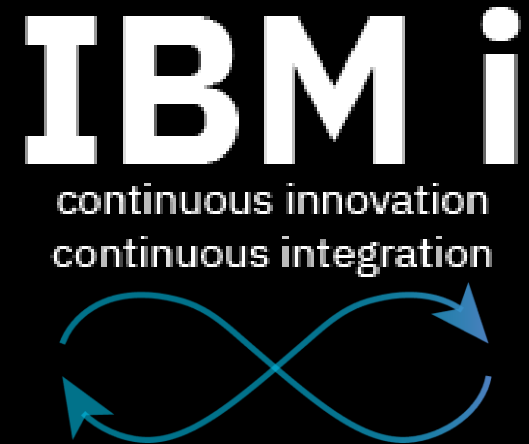
- Sydney (2)
- Osaka
- Tokyo
- Chennai



**NOTE:** IBM PowerVS are physically located in IBM Cloud data centers with their own dedicated infrastructure. For latest list of supported data centers, please visit this [documentation page](#).

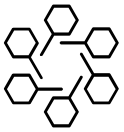
# AI on IBM Power

## – RPG Code Assistant Preview

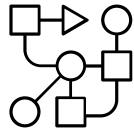


# Code Assistant for IBM i Base Features

A Code Assistant for IBM i tool should help programmers work with existing RPG



**Explain** existing code

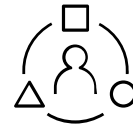


**Generate** modern free-format ILE RPG based on a description

16



**Write** test programs for RPG



... and, if possible, **transform** older RPG into modern, ILE-based free-format



# Get involved! Find out more!

Would you like  
to *get involved?*

Join us in creating the new RPG code assistant for IBM i.

Engage with our team across various research activities, including but not limited to discussions and workshops to help define the offering, alpha & beta testing of the offering pre-GA, and submitting source code to help train and/or test the RPG code assistant model.

Your engagement helps [shape the future of IBM Power](#).

URL:

<https://ibm.biz/wca4i-interest-survey>

QR code:



*Click on links below*

[IBM Statement of Direction](#)

Follow Steve Will on

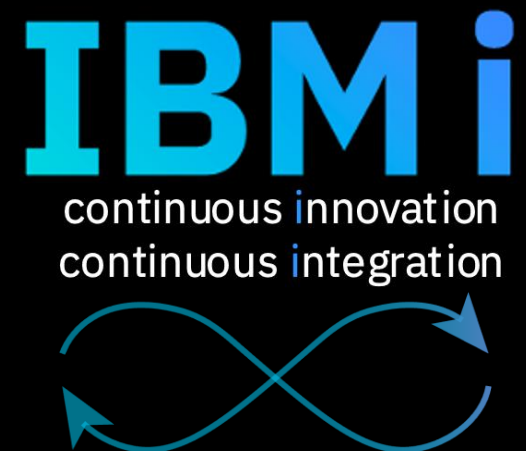
- X (@Steve\_Will\_IBMi) and/or
- LinkedIn  
(<https://www.linkedin.com/in/steve-will-chief-architect-ibm-i/>)
- Tech Channel blog:  
<https://techchannel.com/contributor/steve-will/>

This takes you to a short survey that asks how you would like to be involved. Feel free to forward it to others.



Thank you !

*Demo starts with next chart*

(click in screenshow mode to start)






 RealEstate.rpgle (1-88)  

```
**free
DCL-F   RESTATEDS2   workstn;
DCL-PI  Main  extPgm('REALESTAT2');

END-PI;
/COPY  LAB8,PROTO
agentrate = 5;
EXFMT  SALEINFO;
DOW NOT (*IN03);

    if HousePrice = 0;
```



Please give an overview of the function of this ILE RPG program and the exported procedures. Use 2000 words to do so. Use markdown to highlight variable and subroutine names. Structure the response with whitespace and numbering.

Answer: This is a real estate application that allows users to buy or sell a house by inputting relevant information about the property, such as its price, location, whether it's their first time owning a home, and other fees associated with the transaction. The program presents a screen (SaleInfo) where the user enters this data.

The main processing loop executes until the user decides to exit the program by pressing Function key 3. Within the loop, several checks are performed:

1. If the house price is zero, an error indicator is turned on and the SaleInfo screen is displayed again for the user to correct the input.
2. Similarly, if the residence type or first-time homeowner status fields are left blank, error indicators are activated and the SaleInfo screen is shown for correction.
3. If Function key 6 is pressed, fixed costs for legal fees, survey fees, title insurance, and inspection fees are set. These values are then displayed on the SaleInfo screen.

If none of these conditions are met, the program proceeds to calculate the total cost of the property, which includes legal fees, survey fees, title insurance, and inspection fees. Depending on whether the user is a first-time homebuyer or not, the land tax is calculated differently. The commission rate is applied to the house price to determine the seller's proceeds from the sale.

Additionally, if the total cost of the property is below \$1000,

# The Prompt (the request) drives the output

☰ RealEstate.rpgle (1-88) ✎ ✕

```
**free
DCL-F   RESTATEDS2   workstn;
DCL-PI  Main  extPgm('REALESTAT2');

END-PI;
/COPY  LAB8,PROTO
agentrate = 5;
EXFMT  SALEINFO;
DOW NOT (*IN03);
```



As an ILE RPG and DSPF DDS expert please given an overview of the function of this ILE RPG program and logic flow between screens. Use 2000 words to do so. Use markdown to highlight variables, indicators and subroutine names. Explain the usage and meaning of indicators. Structure the response with whitespace and numbering.

Autodetect (1) - ai4rpg\_241007:latest @

⌘ ↵ @codebase ↵ Enter

Answer: The ILE RPG program is designed for a real estate transaction between a buyer and a seller. It begins by declaring the display file `RESTATEDS2` as a workstation file, which will be used to interact with the user through display formats (DFMs). The main procedure of the program is encapsulated in the `Main` procedure interface, which indicates that it is an external program called 'REALESTAT2'.

The `Main` procedure contains a `DOW` (Do-While) loop that repeatedly executes its body while the indicator `*IN03` not on. This suggests that the program has multiple screens or stages of interaction with the user, as each iteration represents a new screen display.

Within the loop:

1. The program checks if certain required fields are empty (i.e., House Price, Residence type, First-time homebuyer status) and sets on corresponding indicators ( `*IN81` , `*IN82` , `*IN83` ) to prompt the user to fill them out before proceeding. If any of these fields are not filled out, the program uses `EXFMT` (Write/Then Read Format) to write the SaleInfo format to the display and then waits for the user input with `READ` . The `ITER` statement is used to continue the loop, prompting the user again for the missing information.
2. If all required fields contain data, it executes two subroutines: `BuyerSr` and `SellerSr` . These subroutines calculate

5. After all calculations, the SaleInfo format is written to the display with `EXFMT`, and if indicator `*IN79` is on, it means a result format is also ready to be displayed, prompting the user to review the details of their real estate transaction.
6. The program checks if the exit indicator `*IN03` is off. If it isn't, it runs another subroutine named `clear`, which resets all transaction-related variables to their initial state or to a neutral value indicating no data entered (e.g., 0 for numeric values and empty space for character values).
7. The program uses another `EXFMT SaleInfo` statement to redisplay the SaleInfo screen, preparing for the next iteration of the loop unless the exit condition is met.

The program ends when the user exits the loop (i.e., sets `*IN03` on), and it sets on the `*INLR` indicator to end the job and returns control back to the system.

In summary, this ILE RPG program is used for buying and selling houses by interacting with users through display files. It guides users through a series of screens where they input necessary data such as house price, residence status, first-time homebuyer designation, and conducts calculations to determine costs and proceeds from the sale based on various fees and taxes. The program also allows users to clear entered data and exit at any point during the transaction.



# Do you want to provide material to help us train the model?

- If you want to submit code, let us know you want to help
  - e-mail [AlforIBMi@ibm.com](mailto:AlforIBMi@ibm.com)
- Agree to the license
  - We'll send it to you – it says we can use your submitted material to train the model
- Submit code
  - Information on how is found at: <https://ibm.github.io/rpg-genai-data/#/>
  - You decide if others (outside IBM) can see & use your code or not
- At some point in the future, we will likely ask for volunteers to actually use the submissions in training & evaluating

