東京 2025
LINUX PLUMBERS CONFERENCE

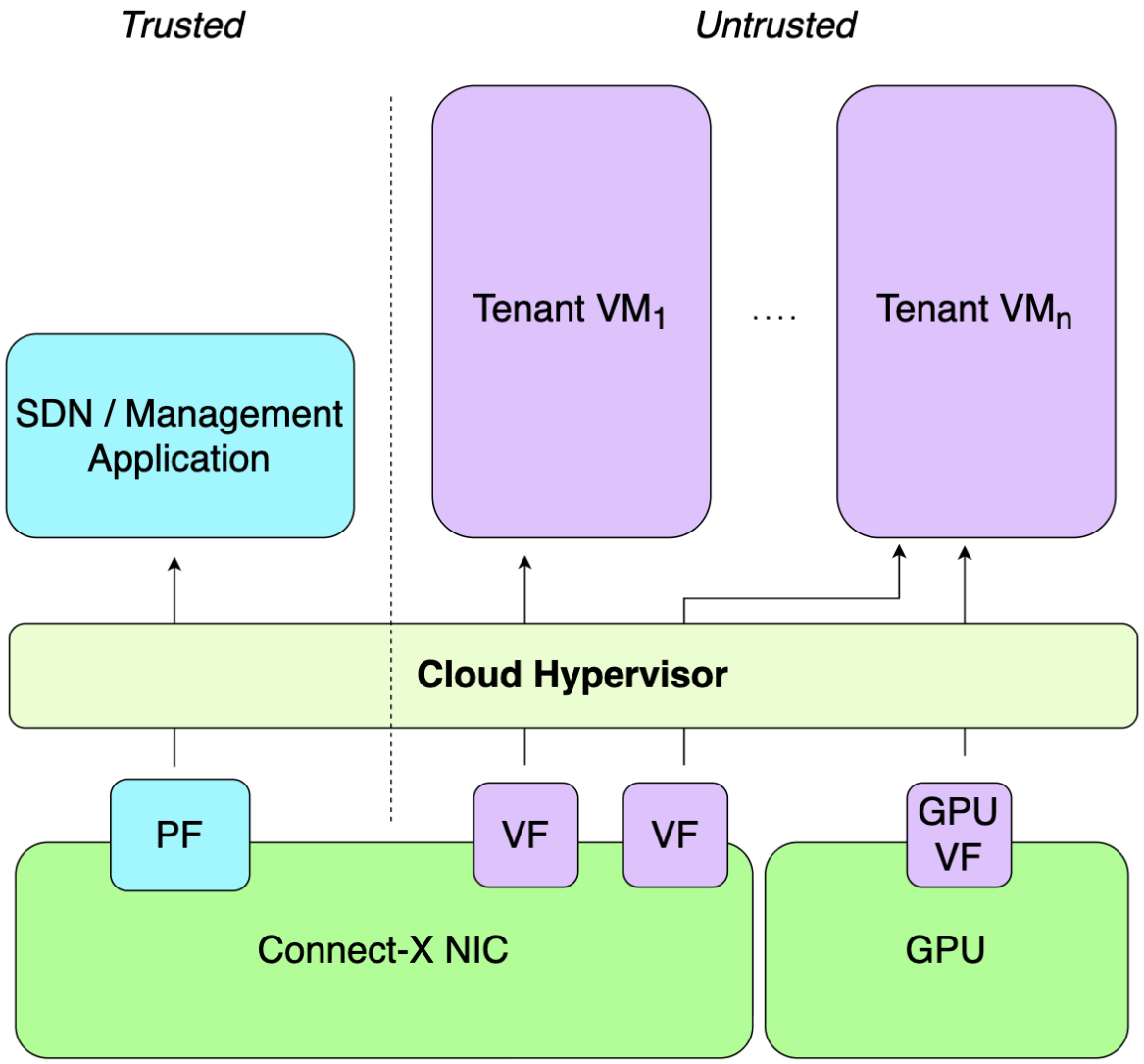
TOKYO, JAPAN / DECEMBER 11-13, 2025

Supporting Hypervisor Kexec with Modern Devices

Adithya Jayachandran, Saeed Mahameed

Generic Cloud Architecture

- Considers tenant as untrusted
- Requires tenants to access physical hardware (GPUs, network cards, storage)
- Management and virtualization stacks run on hypervisors
- SDN manages routing and device access of VMs



Requirements for High Availability

Host software updates are disruptive for modern NICs

- Preserve tenant VM state without interruptions
  - Tenants should be able to continue running workloads without noticing
- Application state above PCI should be gracefully maintained
  - Kexec can restore PCI state but user-space is abruptly killed
- Updates for management software stack should happen atomically

Naïve Solution

- Kexec with full driver support of KHO
  - On suspend, pause all memory operations and store memory in persistent memory
  - On resume, restore from memory

Naïve Solution

- Kexec with full driver support of KHO
  - On suspend, pause all memory operations and store memory in persistent memory
  - On resume, restore from memory

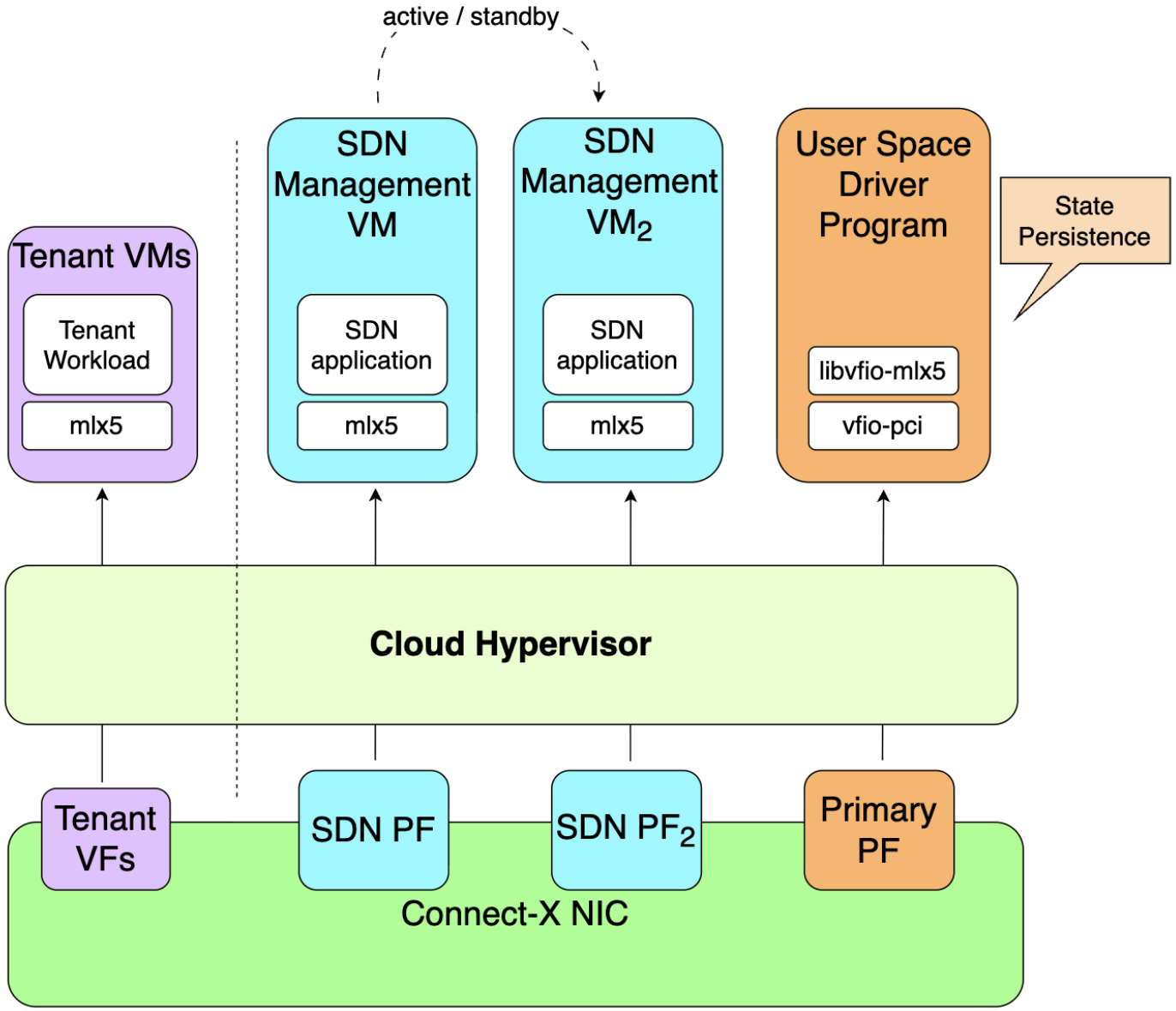
Too complex for some devices...

Complexity with Modern Devices

- Modern drivers manage the whole device state in memory
- Variety of software stacks to serialize/restore
  - More than one driver will be in use by a device
  - Software layers need to be adapted
  - RDMA devices have multiple users with minimal kernel intervention
    - Millions of objects **per user**
- Same device used across multiple tenants
  - SR-IOV

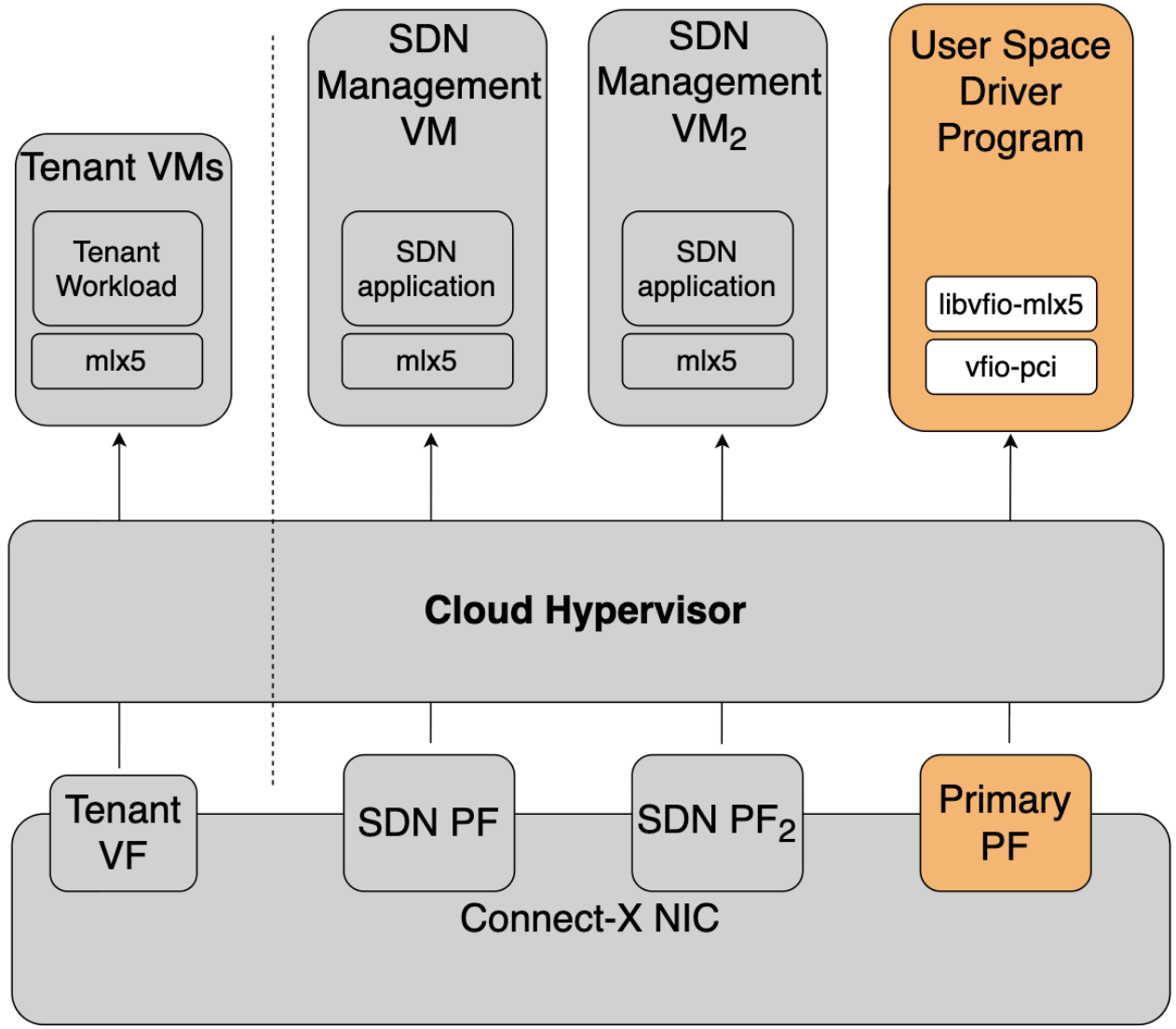
Solution Architecture

- VFs managed by user space driver program
- Two SDN VMs (1 active, 1 standby)
- Tenant VMs run as usual



State Preservation (vfio-mlx5)

- vfio-pci user space process to drive a PF
- Memory manager for the tenant VFs
  - Can point to a KHO folio or memfd to store state in kexec persistent memory
- Kexec suspend/resume aware



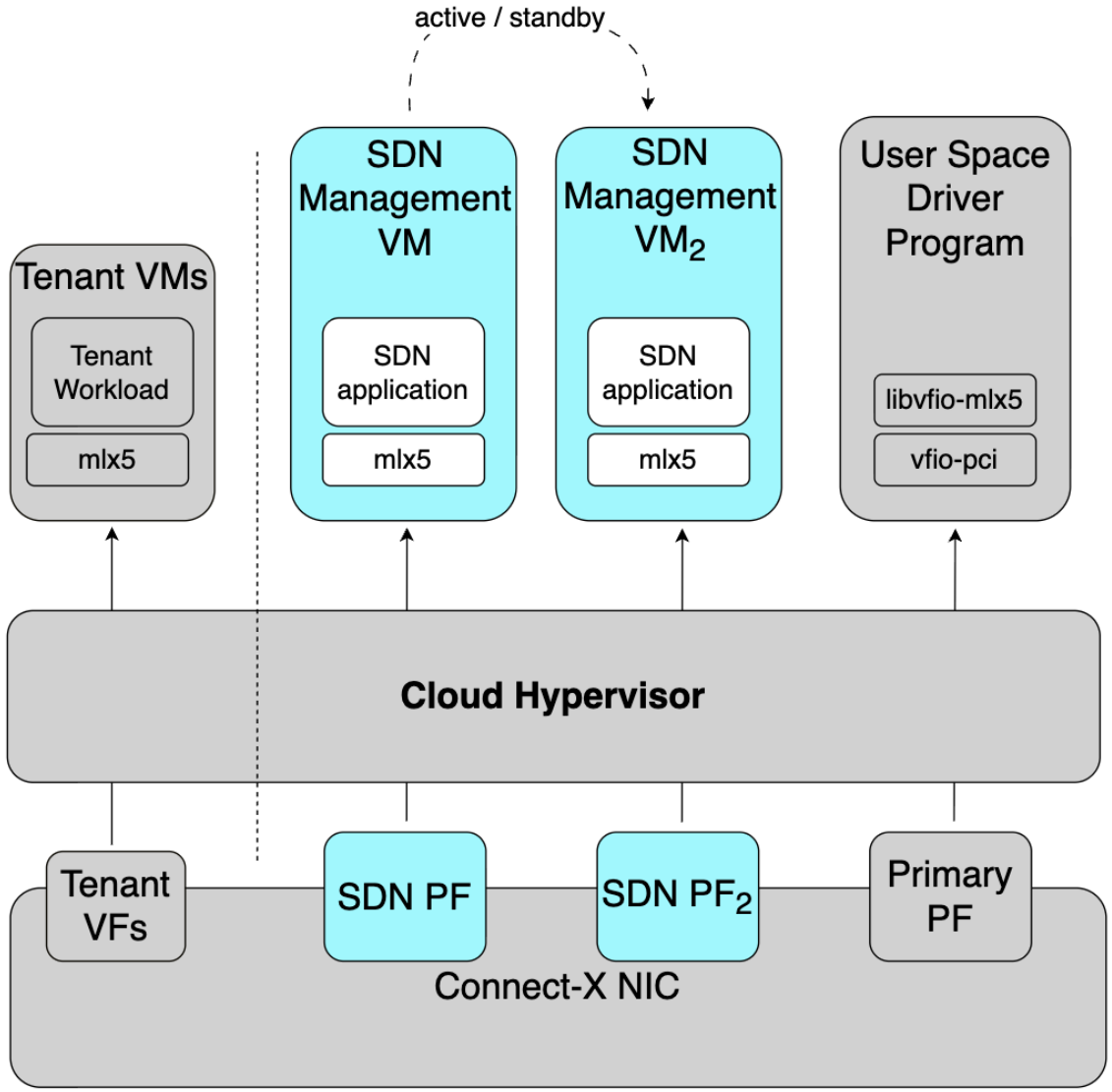
State Preservation (vfio-mlx5)

```
Usage: vfio_mlx5 [OPTIONS]
Options:
  --help                Show this help message
  --device=PCIBDF, -d PCIBDF   Add PCI device (can be specified up to 8 times)
          use PCIBDF,vf_token=... to add PCI device with vf_token
  --memsize=SIZE, -m SIZE      Set memory size in bytes (4K aligned, default 128MB)
  --nvfs=NUM, -n NUM           Set number of vfs for devices (default 0)
  --file=FILE, -f FILE         Log output to file (default stdout)
  --stats=int, -s int          Set stats collection interval in seconds (default 1)
  --noiommu                    Enable NoIOMMU mode
  --iova=IOVA_ADDR, -i IOVA_ADDR  Set IOVA start address (default 0x400000000)

Example:
  vfio_mlx5 --device=0000:03:00.0 --memsize=256M --nvfs=2
  vfio_mlx5 --device=0000:03:00.0 --memsize=1g --nvfs=2 --noiommu
```

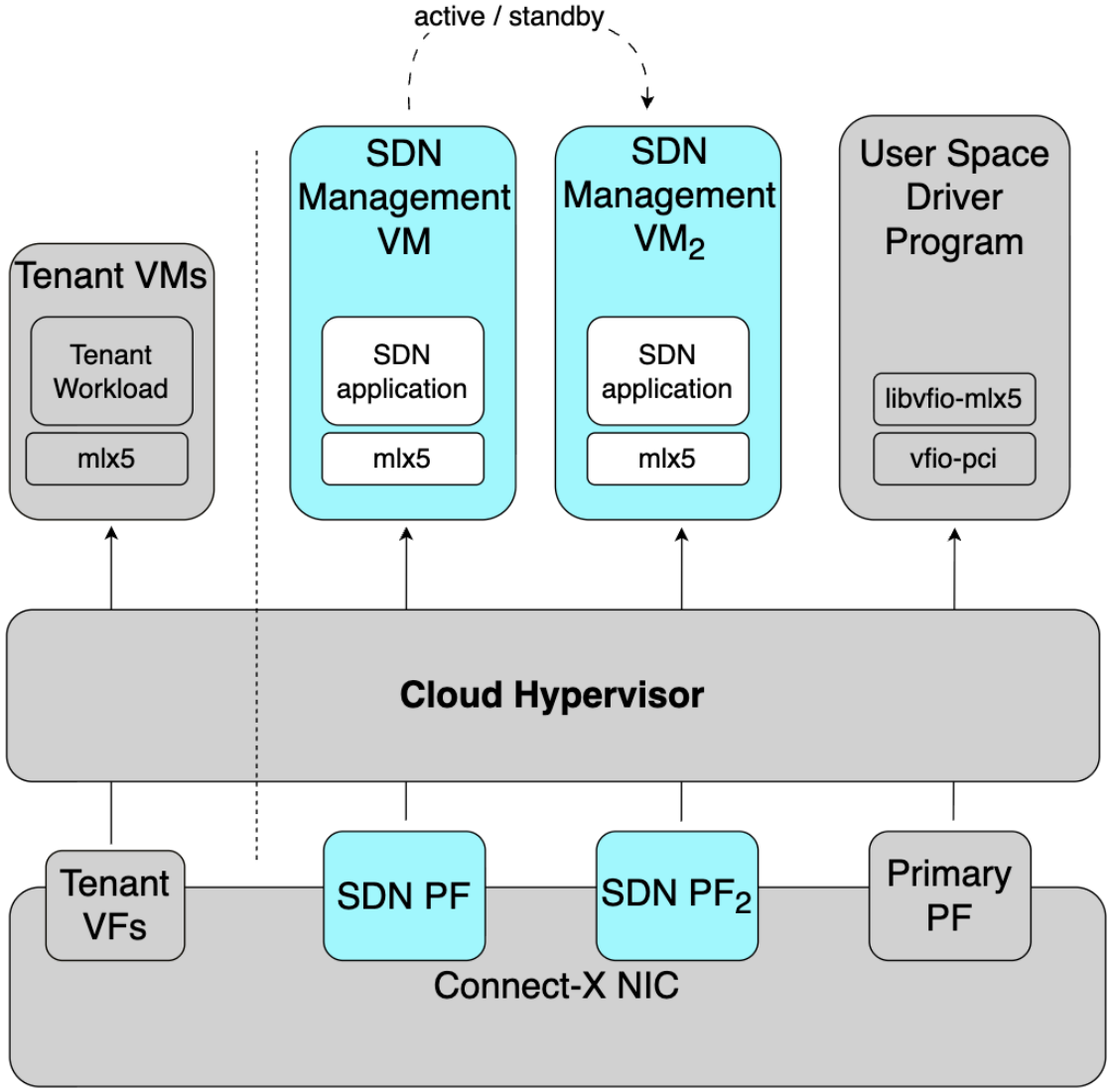
SDN in a VM

- SDN / Management runs in a VM
- During offload enablement, SDN discovers and begins managing the tenant VF's traffic
- No state stored, boot another VM and VF attaches to pipeline



SDN VM Active/Backup Switchover

- How can we update an SDN software stack?
  - Start a parallel SDN VM that is updated but inactive
  - Once ready, activate the new pipeline (eswitch)
- Tenant notices no downtime when switching pipelines
- Can be done outside of kexec



Links

- [1] mlx5 driver support for vport discovery
- [2] [3] devlink switchdev_inactive mode and kernel changes
- [4] libvfio_mlx5 library with vfio_mlx5 sample application

東京 2025
LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025