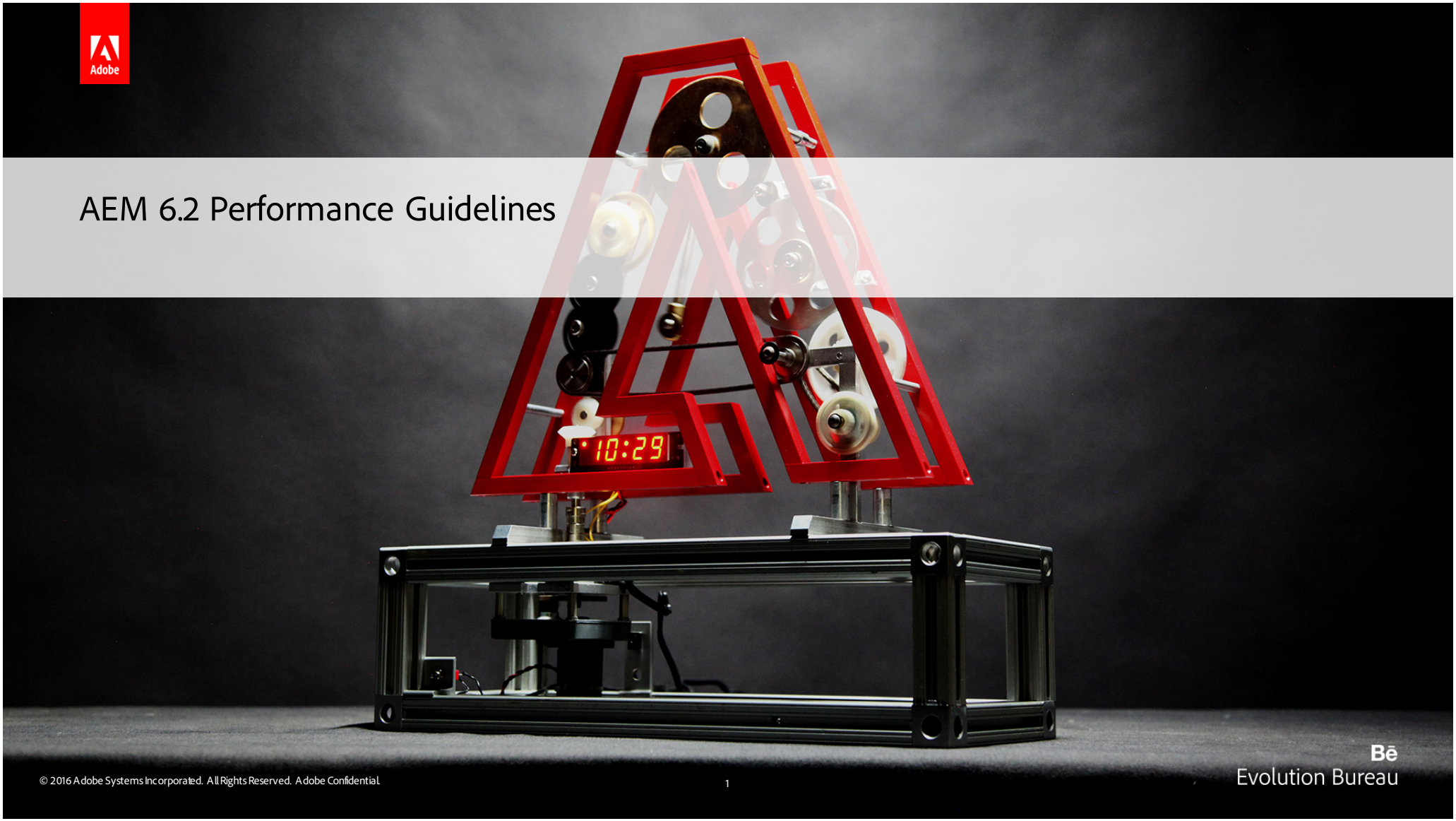
AEM 6.2 Performance Guidelines



Evolution Bureau

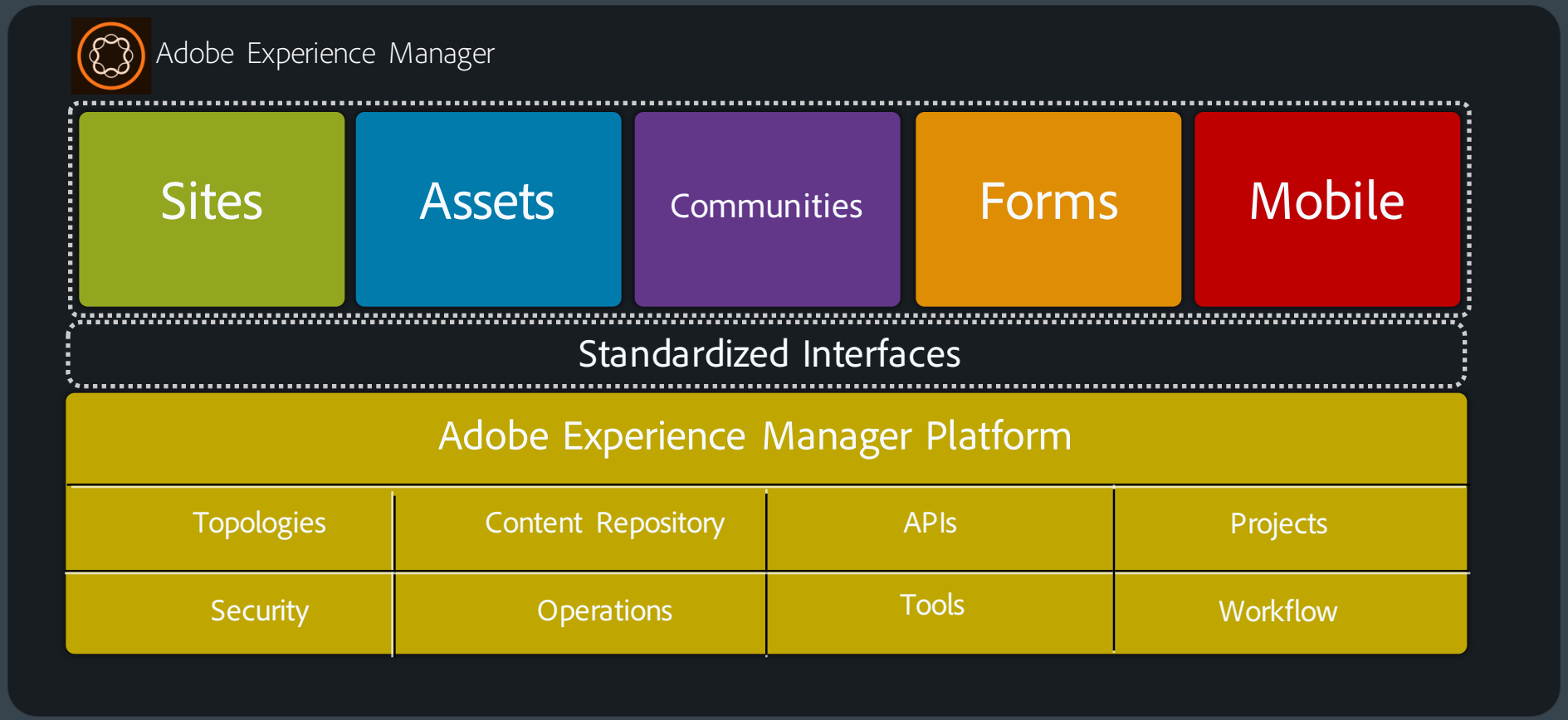
Table of Content

When to use these Performance Guidelines?

1) **First time deployment:** When planning to deploy AEM Sites or Assets for the first time, it's important to understand the options available for configuring the Micro Kernel, Nodestore, and Datastore vs the default settings
   - Example: changing the default of the Datastore forTarMK to File Datastore
2) **Upgrading to a new version**: When upgrading to a new version, it's important to understand the performance differences compared to the running environment.
   - Example: upgrading from 6.1 to 6.2, or from 6.0 CRX2 to 6.2 OAK.
3) **Response time is slow**: When the selected Nodestore architecture is not meeting your requirements, it's important to understand the performance differences compared to other topology options.
   - Example: deploying TarMK instead of MongoMK, or using a File or S3 Shared Datastore
4) **Adding more authors**: when the recommended TarMK topology is not meeting the performance requirements and upsizing the Author node has reached the maximum capacity available, it's important to understand the performance differences compared to using MongoMK with 3 or more Author nodes
   - Example: deploying MongoMK instead of TarMK
5) **Adding more content**: When the recommended DataStore architecture is not meeting your requirements, it's important to understand the performance differences compared to other Datastore options.
   - Example: using the S3 Datastore instead of a File Datastore

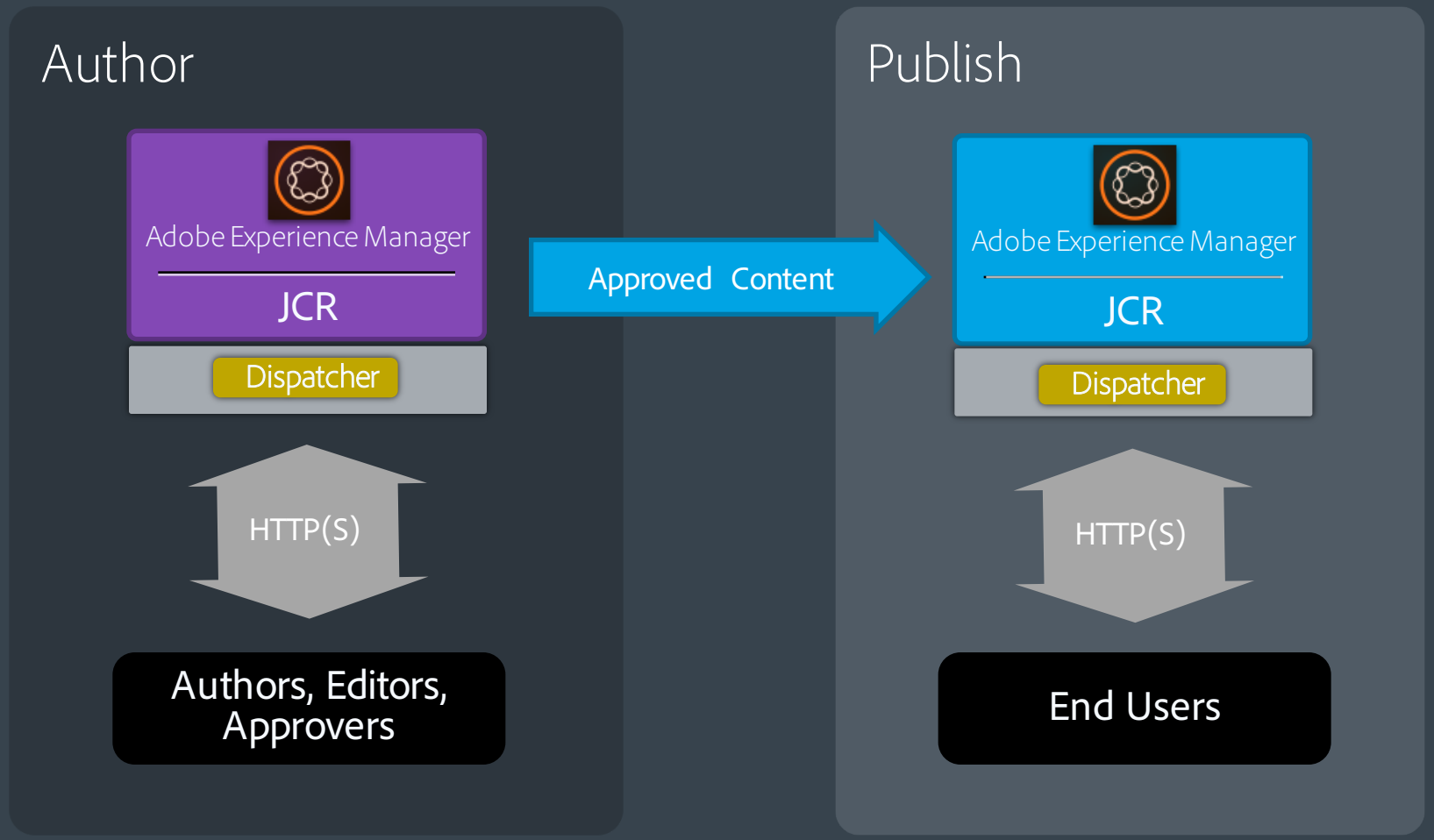
Table of Content

Introduction: AEM Platform



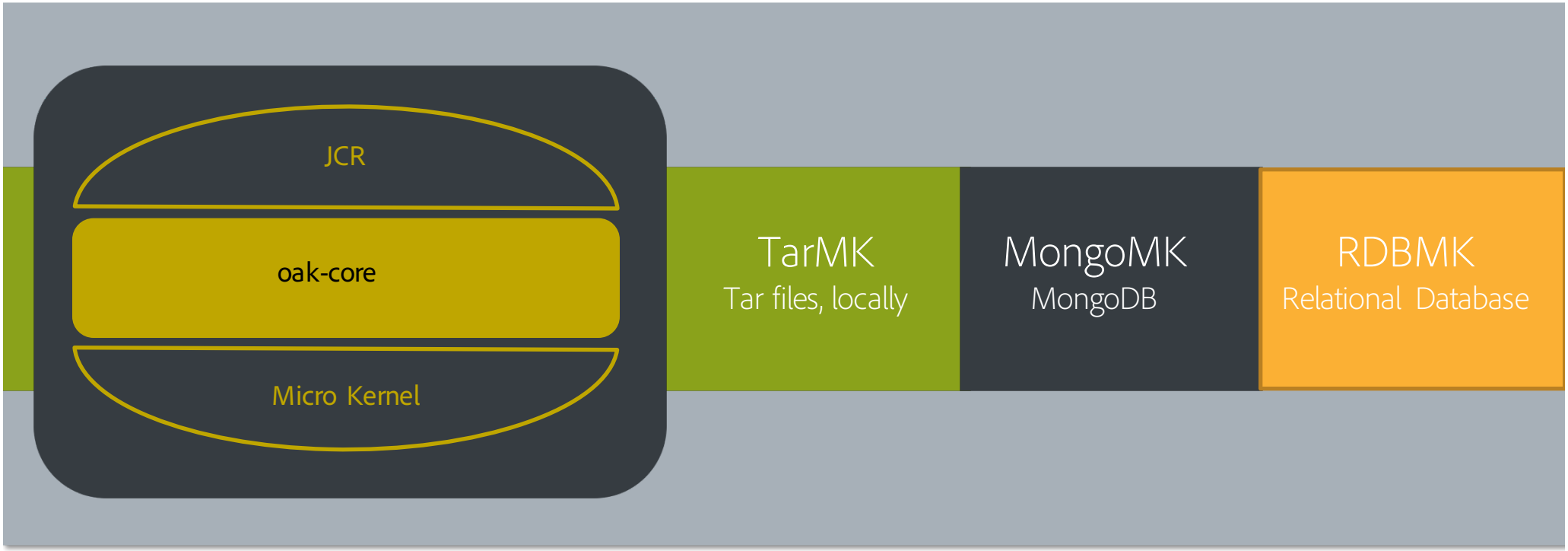
Adobe

Introduction: AEM Architecture

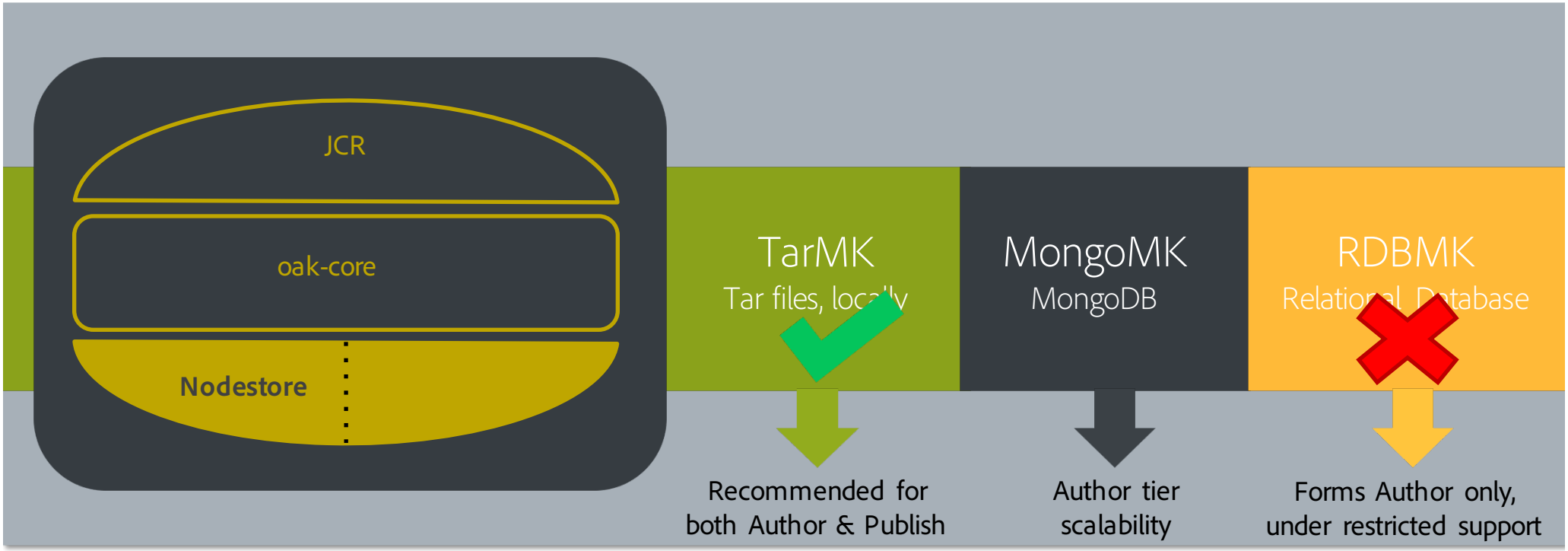


Adobe

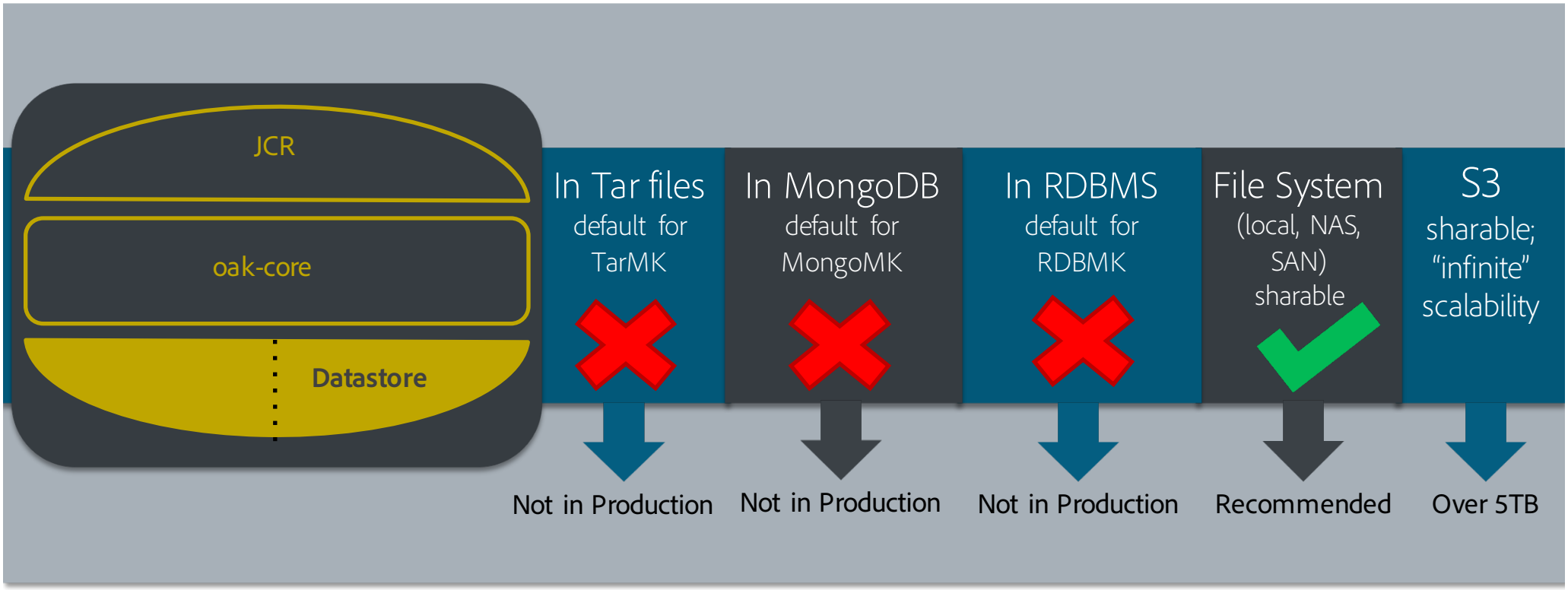
Micro Kernels – Where JCR persists its content and state



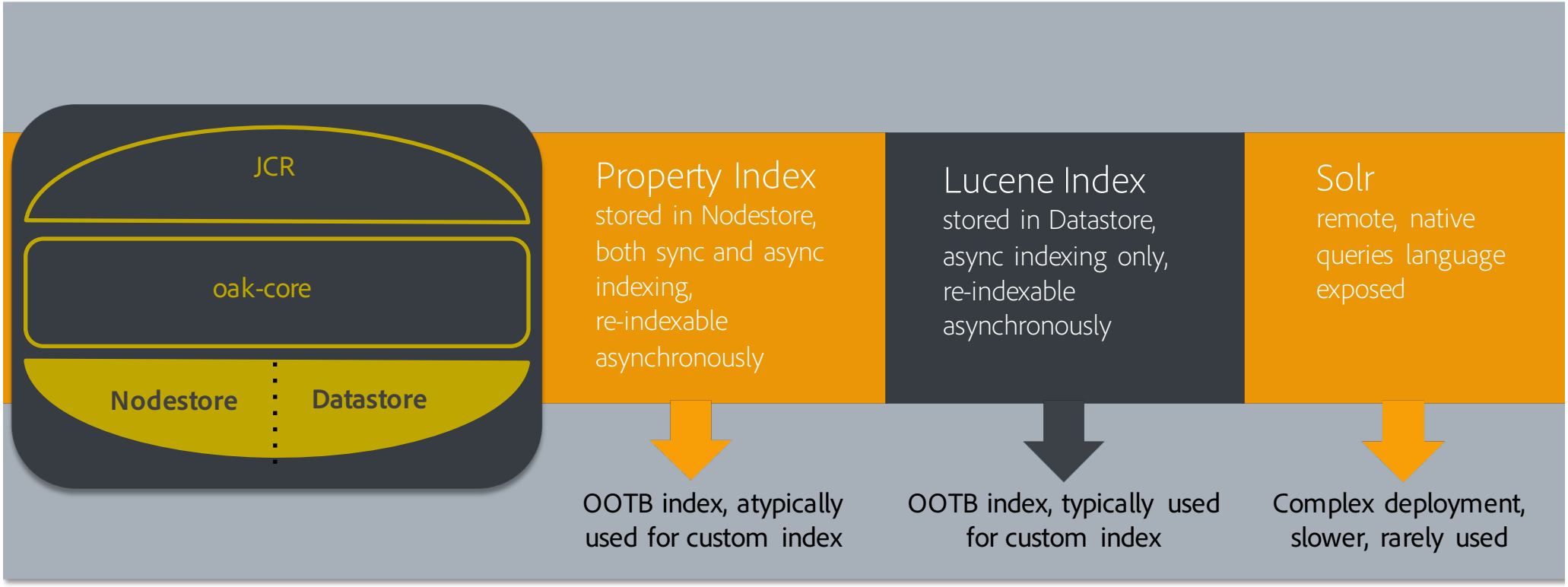
Node Store – Where nodes and properties get stored



Data Stores – Where the large blobs get stored



Search – Pluggable Index Providers



Develop… maintainable, reusable & upgradable code

- Bring in AEM certified developers & architects
- Share coding conventions with review process
- Well defined versioning and branching patterns
- Separation of presentation, logic, and content
- Stick to AEM or Sling APIs
- Recommended tooling: AEM Eclipse plugin, Bracket, Maven
- Develop in the context of actual content

- Create development silos
- Use JCR APIs directly, if you can
- Go around standard APIs (/etc, /var)
- Change /libs, but rather use overlays
- Change /apps/geometrix directly, copy it over to /apps/myproject
- Make OSGI manual changes
- Make CRX/DE manual changes

Introduction: Benchmark scenarios

A. **Single Product scenario:**

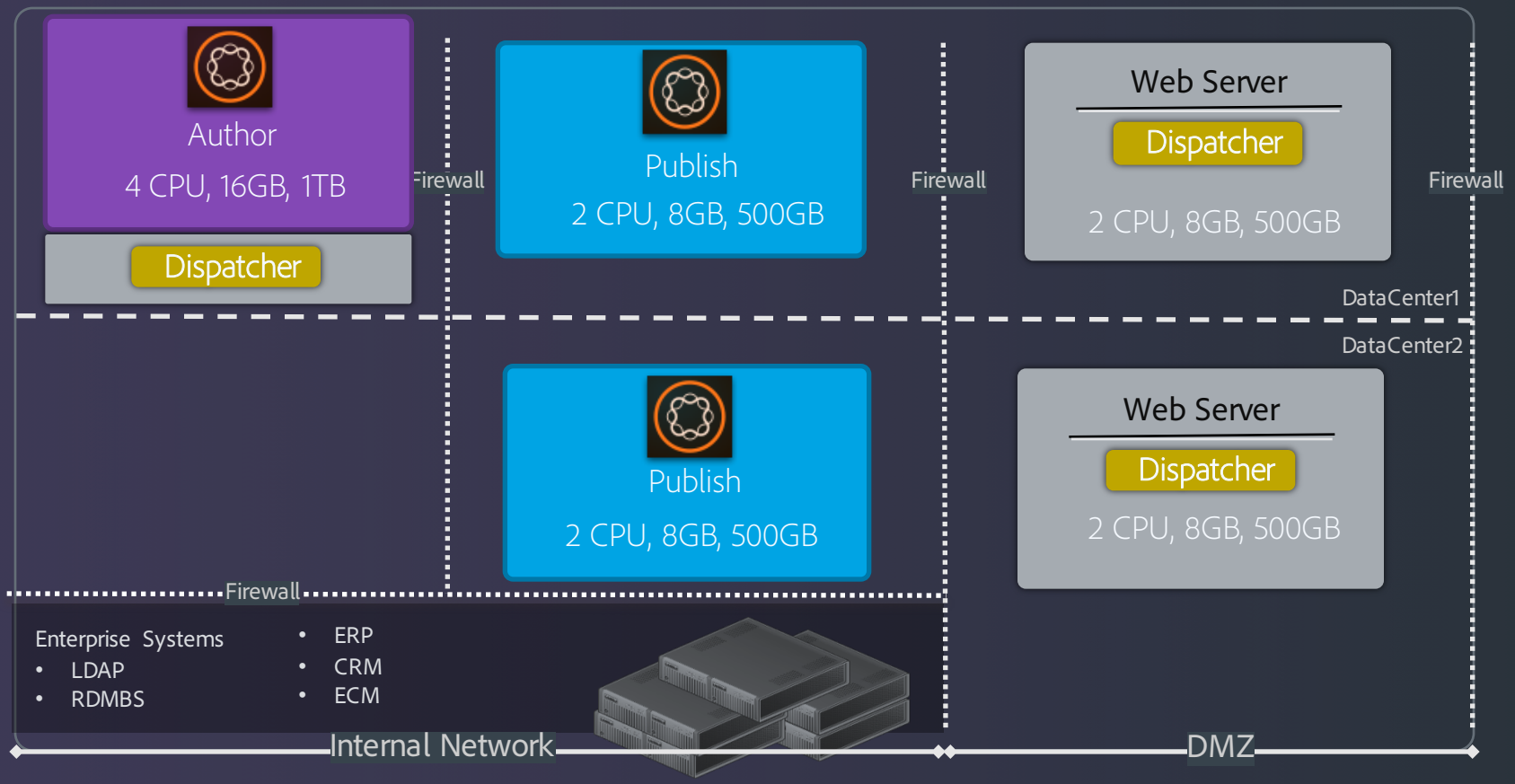
- AEM Assets:
  - User interactions: Browse Assets / Search Assets / Download Asset / Read Asset Metadata / Update Asset Metadata / Upload Asset / Run Upload Asset Workflow
  - Execution mode: concurrent users, single interaction per user

B. **Mix Products scenario:**

- AEM Sites + Assets:
  - Sites user interactions: Read Article Page / Read Page / Create Paragraph / Edit Paragraph / Create Content Page / Activate Content Page / Author Search
  - Assets user interactions: Browse Assets / Search Assets / Download Asset / Read Asset Metadata / Update Asset Metadata / Upload Asset / Run Upload Asset Workflow
- Execution mode: concurrent users, mix interactions per user

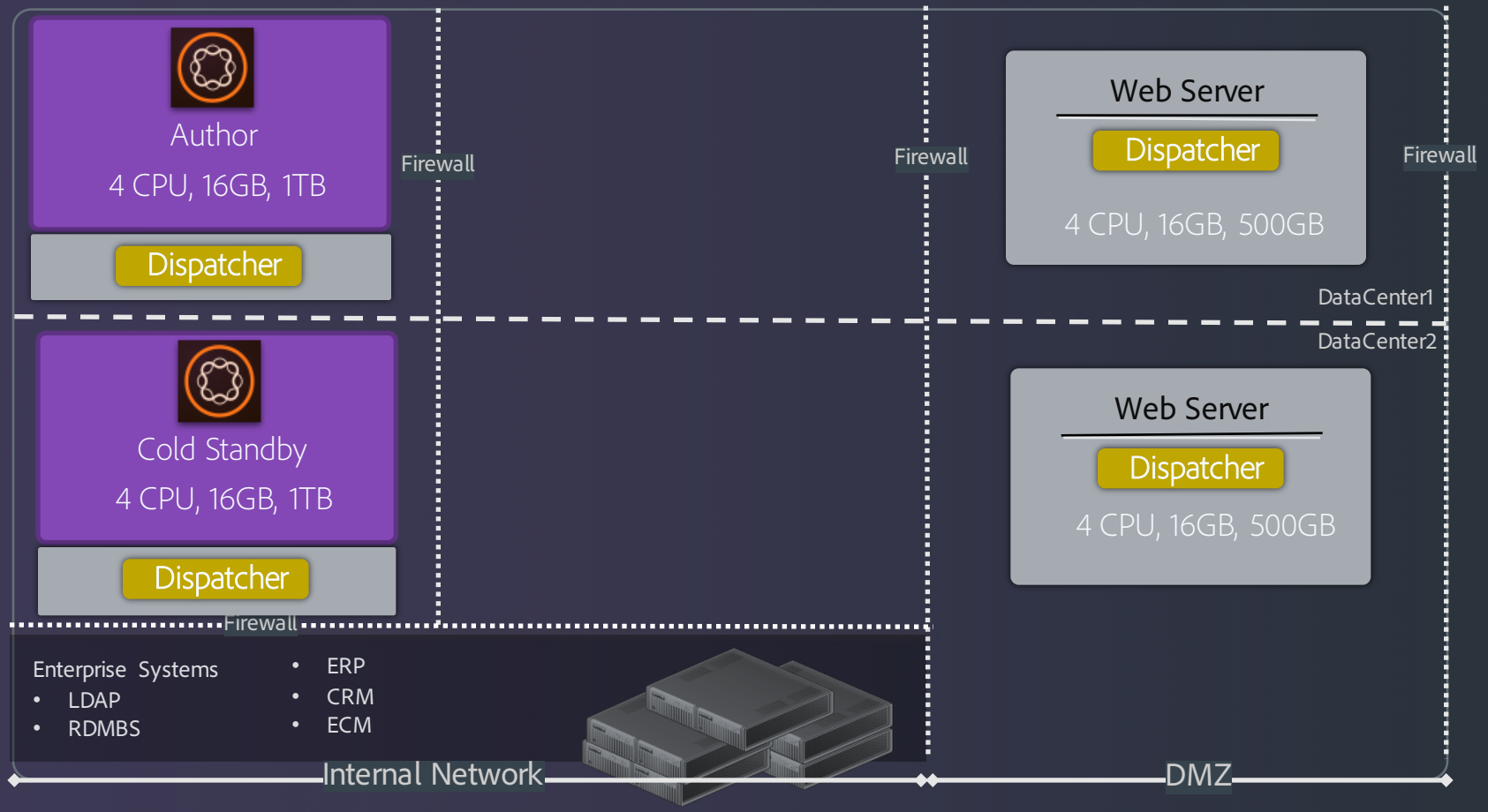
Table of Content

TarMK: Minimum Architecture Guidelines for AEM Sites



Adobe

TarMK: Minimum Architecture Guidelines for AEM Assets



Adobe

TarMK: Settings Guidelines

Setting	Parameter	Value	Description
Sling Job Queues	queue.maxparallel	Set value to half of the # of CPU cores.	By default the number of concurrent threads per job queue is equal to the number of CPU cores.
Granite Transient Workflow Queue	Max Parallel	Set value to half of the # of CPU cores	
JVM parameters	Doak.queryLimitInMemory Doak.queryLimitReads Dupdate.limit Doak.fastQuerySize	500000 100000 250000 True	Add these JVM parameters in the AEM start script to prevent expansive queries from overloading the systems.
Lucene index configuration	CopyOnRead CopyOnWrite Prefetch Index Files	Enabled Enabled Enabled	For more info on available parameters, see the documentation
Data Store = File Datastore	maxCachedBinarySize cacheSizeInMB	1048576 (1MB) or smaller 2-10% of max heap size	For detailed instructions, see the documentation
DAM Update Asset workflow	Transient Workflow	checked	This workflow manages the update of assets
DAM MetaData Writeback	Transient Workflow	checked	This workflow manages XMP write-back to the original binary and sets the last modified date in JCR

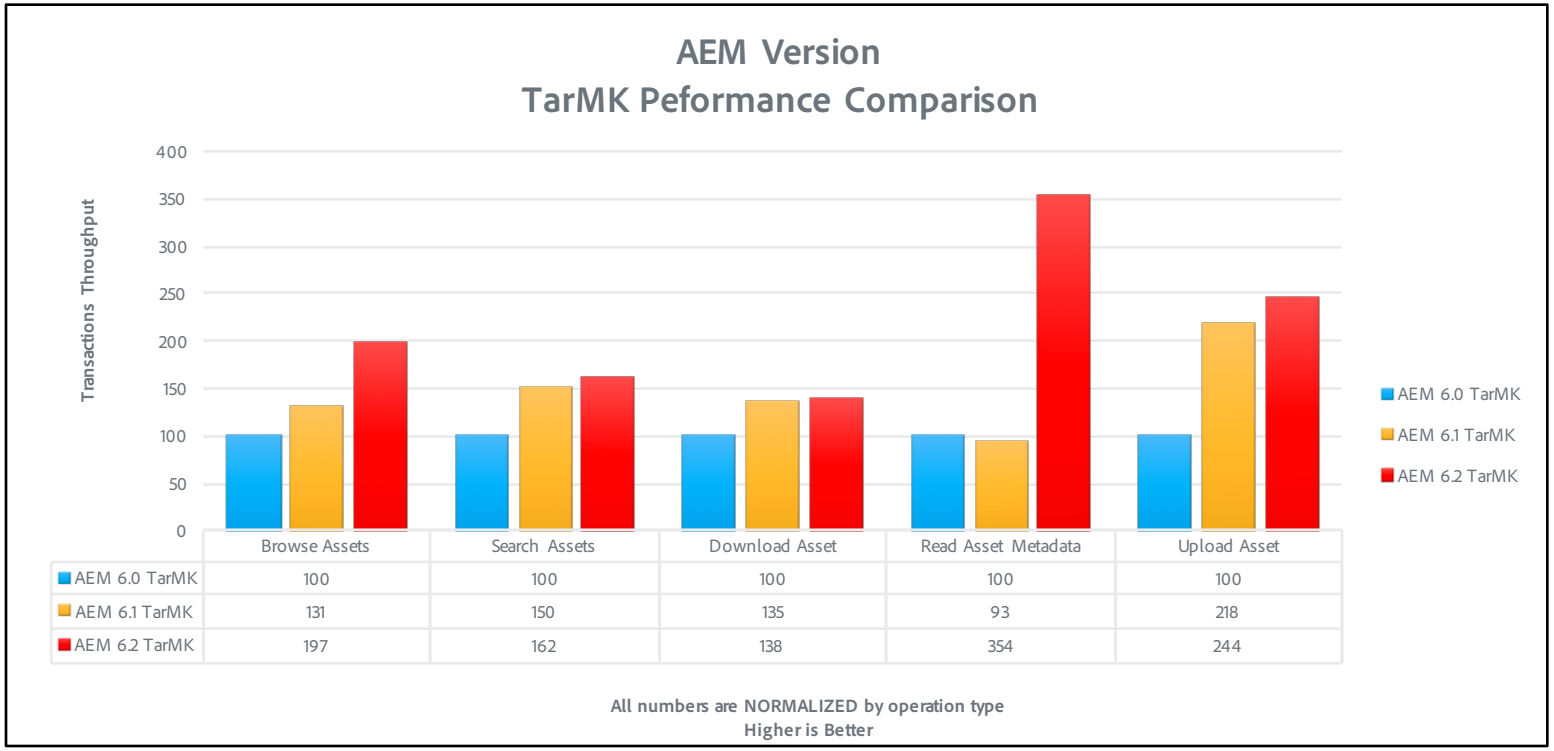
For mode details, please visit: https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html

TarMK: Benchmarks Technical Specifications

	Author node
Server	Bare metal hardware (HP)
Operating System	RedHat Linux
CPU / Cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores
RAM	32GB
Disk	
Java	Oracle JRE Version 8
JVM Heap	16GB
Product	AEM 6.2
Nodestore	TarMK
Datastore	File DS
Scenario	Single Product: Assets / 30 concurrent threads

TarMK: Performance Benchmarks Results

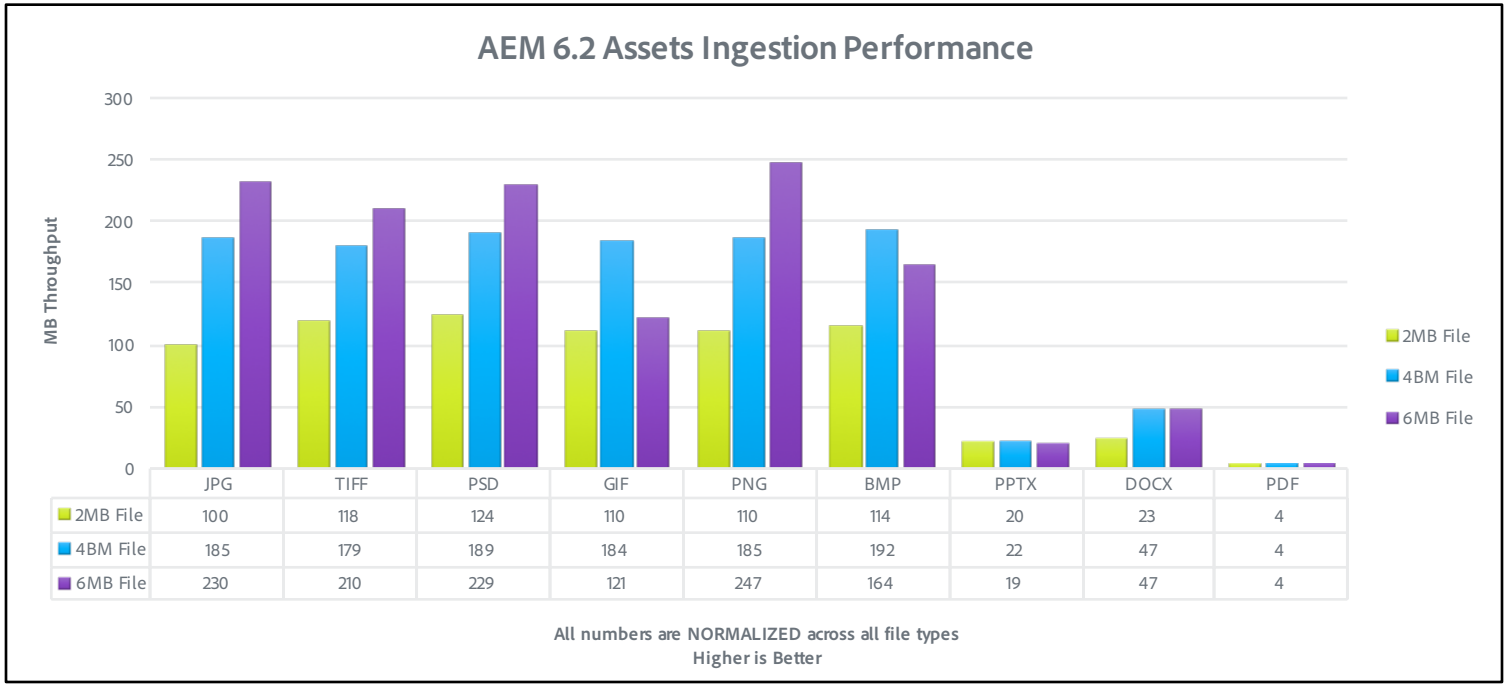
- Note: the numbers below are not actual throughput numbers, they have been normalized to 100 as the baseline



	Browse Assets	Search Assets	Download Asset	Read Asset Metadata	Upload Asset
AEM 6.0 TarMK	100	100	100	100	100
AEM 6.1 TarMK	131	150	135	93	218
AEM 6.2 TarMK	197	162	138	354	244

TarMK: Performance Benchmarks Results

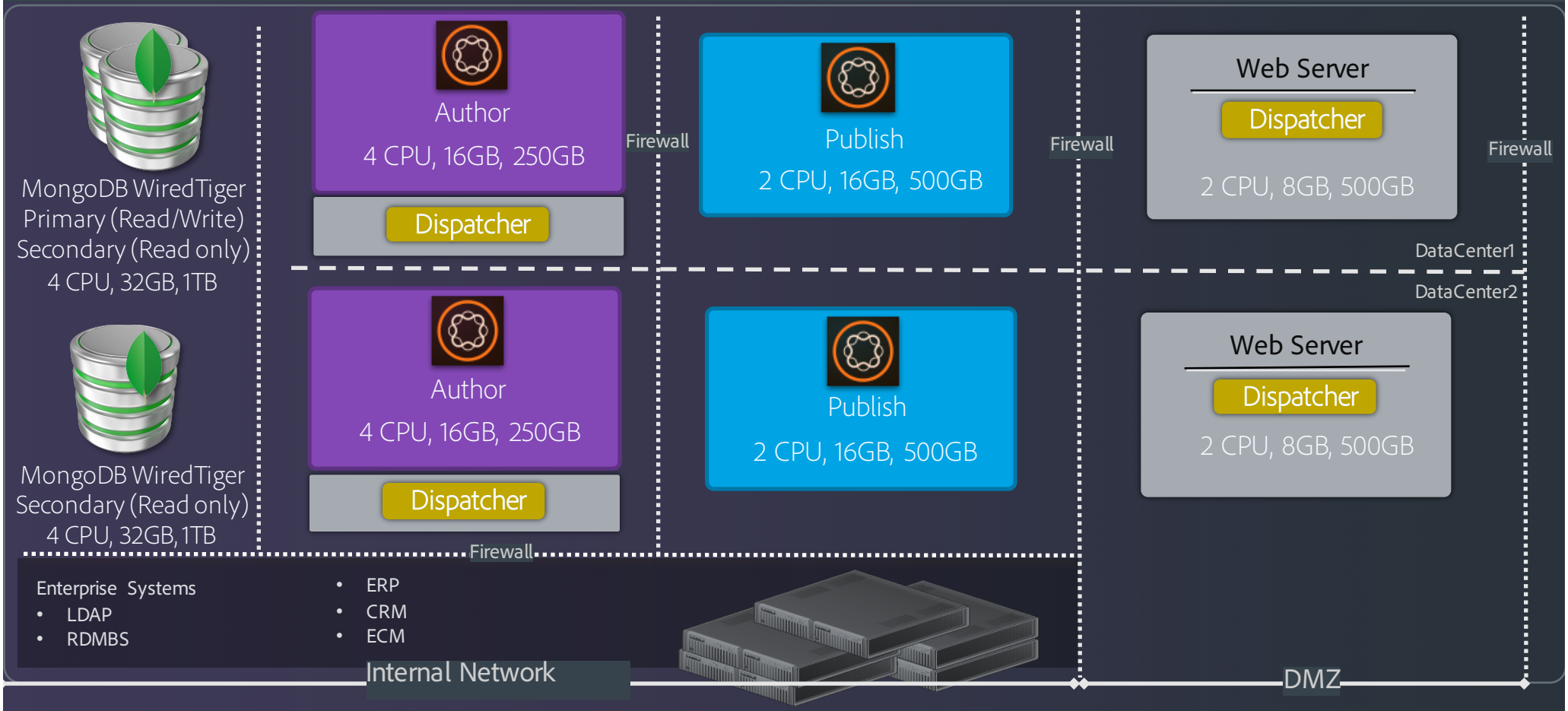
- Note: the numbers below are not actual throughput numbers, they have been normalized to 100 as the baseline



	JPG	TIFF	PSD	GIF	PNG	BMP	PPTX	DOCX	PDF
2MB File	100	118	124	110	110	114	20	23	4
4BM File	185	179	189	184	185	192	22	47	4
6MB File	230	210	229	121	247	164	19	47	4

Table of Content

MongoMK: Minimum Architecture Guidelines for AEM Sites



Adobe

MongoMK: Settings Guidelines

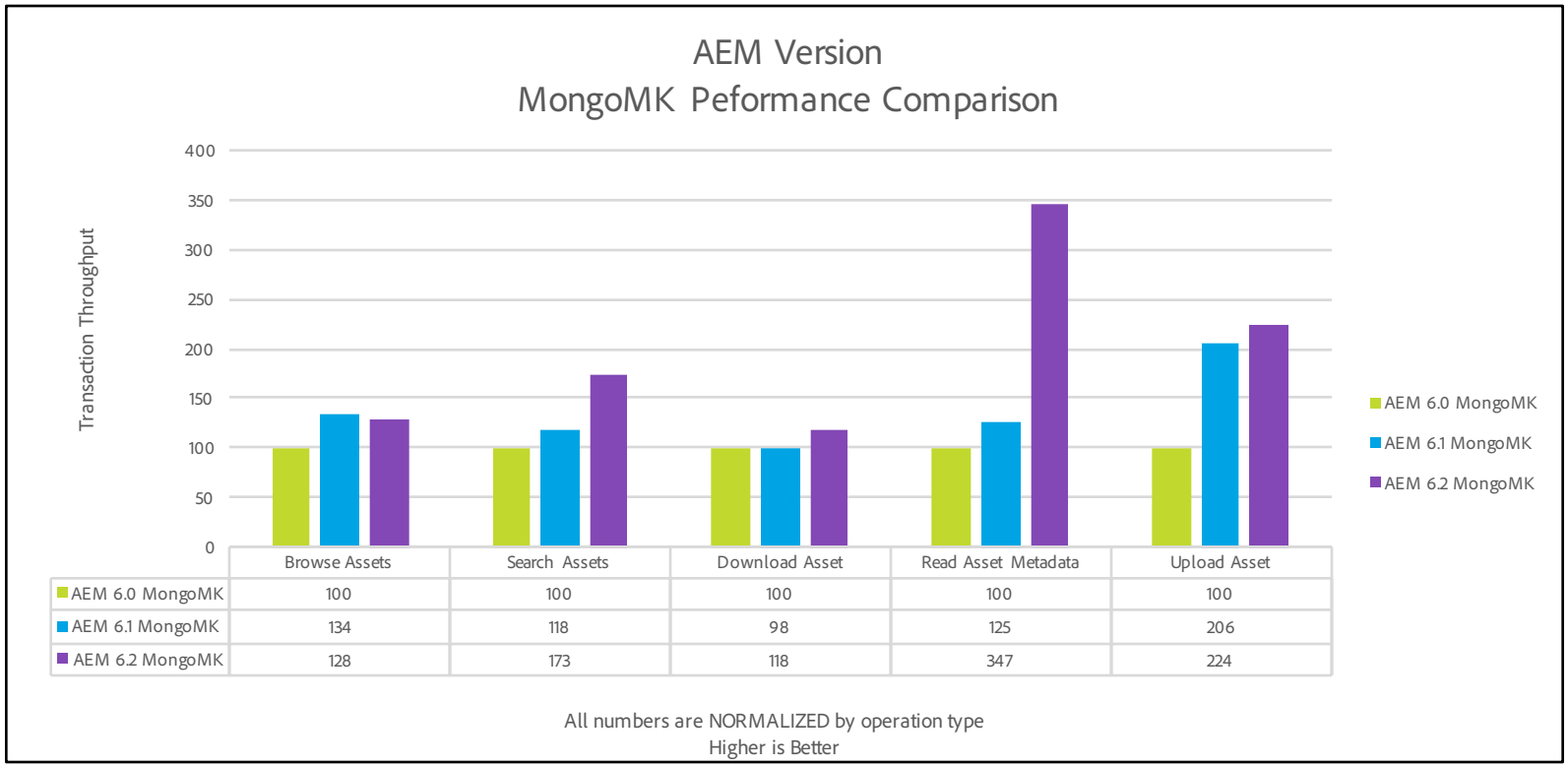
Setting	Parameter	Value	Description
Sling Job Queues	queue.maxparallel	Set value to half of the # of CPU cores.	By default the number of concurrent threads per job queue is equal to the number of CPU cores.
Granite Transient Workflow Queue	Max Parallel	Set value to half of the # of CPU cores	
JVM parameters	Doak.queryLimitInMemory Doak.queryLimitReads Dupdate.limit Doak.fastQuerySize Doak.mongo.maxQueryTimeMS	500000 100000 250000 True 60000	Add these JVM parameters in the AEM start script to prevent expansive queries from overloading the systems.
Lucene index configuration	CopyOnRead CopyOnWrite Prefetch Index Files	Enabled Enabled Enabled	For more info on available parameters, see the documentation
Data Store = File Datastore	maxCachedBinarySize cacheSizeInMB	1048576 (1MB) or smaller 2-10% of max heap size	For detailed instructions, see the documentation
DocumentNodeStoreService	cache nodeCachePercentage childrenCachePercentage diffCachePercentage docChildrenCachePercentagepersistentCache= documentCache prevDocCachePercentage	2048 35 20 30 10 size=2048,binary=0 ~500 MB max or lower 4	The default size of the cache is set to 256 MB Has impact on time to perform cache invalidation Default value
oak-observation	thread pool length	min & max = 20 50000	

MongoMK: Benchmarks Technical Specifications

	Author node	MongoDB node
Server	Bare metal hardware (HP)	Bare metal hardware (HP)
Operating System	RedHat Linux	RedHat Linux
CPU / Cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores
RAM	32GB	32GB
Disk		
Java	Oracle JRE Version 8	N/A
JVM Heap	16GB	N/A
Product	AEM 6.2	MongoDB 3.2 WiredTiger
Nodestore	MongoMK	N/A
Datastore	File DS	N/A
Scenario	Single Product: Assets / 30 concurrent threads	

MongoMK: Performance Benchmarks Results

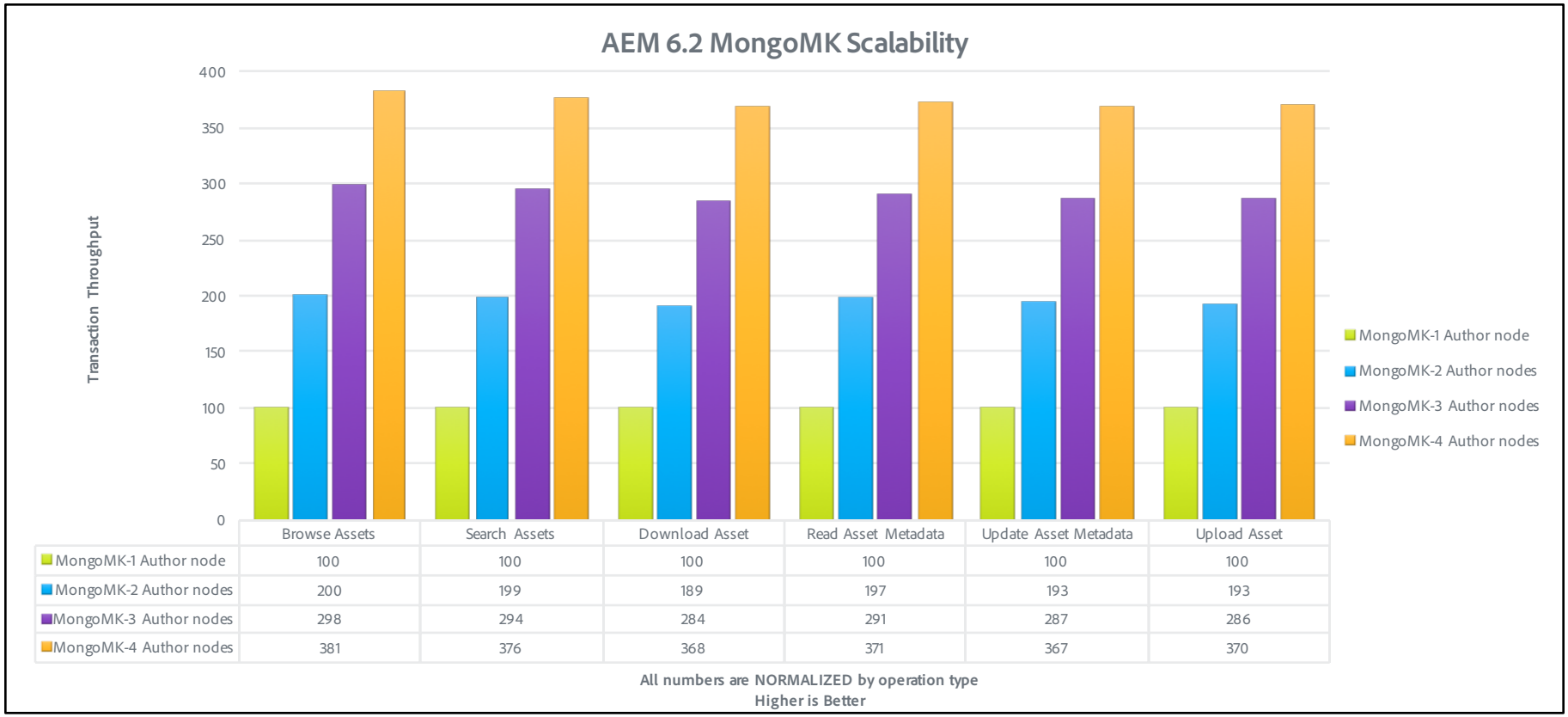
- Note: the numbers below are not actual throughput numbers, they have been normalized to 100 as the baseline



	Browse Assets	Search Assets	Download Asset	Read Asset Metadata	Upload Asset
AEM 6.0 MongoMK	100	100	100	100	100
AEM 6.1 MongoMK	134	118	98	125	206
AEM 6.2 MongoMK	128	173	118	347	224

MongoMK: Performance Benchmarks Results

- Note: the numbers below are not actual throughput numbers, they have been normalized to 100 as the baseline



	Browse Assets	Search Assets	Download Asset	Read Asset Metadata	Update Asset Metadata	Upload Asset
MongoMK-1 Author node	100	100	100	100	100	100
MongoMK-2 Author nodes	200	199	189	197	193	193
MongoMK-3 Author nodes	298	294	284	291	287	286
MongoMK-4 Author nodes	381	376	368	371	367	370

Table of Content

TarMK vs MongoMK: Benchmarks Technical Specifications

	Author CRX2 node	Author OAK node	MongoDB node
Server	Bare metal hardware (HP)	Bare metal hardware (HP)	Bare metal hardware (HP)
Operating System	RedHat Linux	RedHat Linux	RedHat Linux
CPU / Cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores
RAM	32GB	32GB	32GB
Disk			
Java	Oracle JRE Version 7	Oracle JRE Version 8	N/A
JVM Heap	16GB	16GB	N/A
Product	CQ 5.6.1	AEM 6.2	MongoDB 3.2 WiredTiger
Nodestore	TarPM	TarMK or MongoMK	N/A
Datastore	File DS	File DS	N/A
Scenario	Single Product: Assets / 30 concurrent threads per run		

TarMK vs MongoMK: Performance Benchmarks Results

- Note: the numbers below are not actual throughput numbers, they have been normalized to 100 as the baseline

	Browse Assets	Search Assets	Download Asset	Read Asset Metadata	Update Asset Metadata	Upload Asset	Update Asset Workflow
AEM 6.0 CRX2	100	100	100	100	100	100	100
AEM 6.2 TarMK	126	113	143	34	145	94	99
AEM 6.2 MongoMK	112	139	101	30	50	94	99

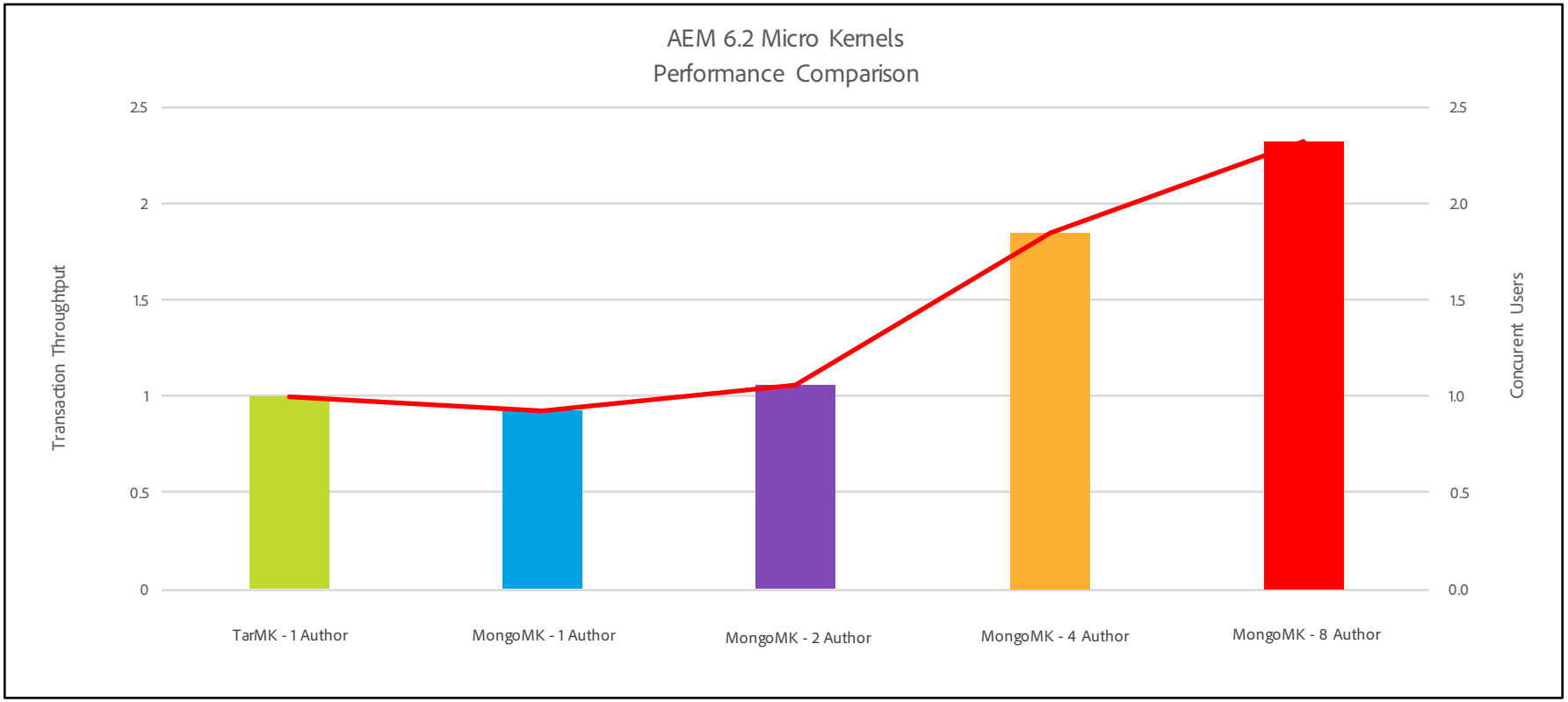
All numbers are NORMALIZED by operation type
Higher is Better

TarMK vs MongoMK: Benchmarks Technical Specifications

	Author TarMK node	Author MongoMK node	MongoDB node
Server	AWS	AWS	AWS
Operating System	RedHat Linux	RedHat Linux	RedHat Linux
CPU / Cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores	Intel(R) Xeon(R) CPU E5-2407 @2.40GHz, 8 cores
RAM	32GB	32GB	32GB
Disk			
Java	Oracle JRE Version 8	Oracle JRE Version 8	N/A
JVM Heap	16GB	16GB	N/A
Product	AEM 6.2	AEM 6.2	MongoDB 3.2 WiredTiger
Nodestore	TarMK	MongoMK	N/A
Datastore	File DS	File DS	N/A
Scenario	Mix Products: Sites+Assets / 200 concurrent threads		

TarMK vs MongoMK: Performance Benchmarks Results

- Note: the numbers below are not actual throughput numbers, they have been normalized to 1 as the baseline



TarMK vs MongoMK Guidelines

Benefits of TarMK

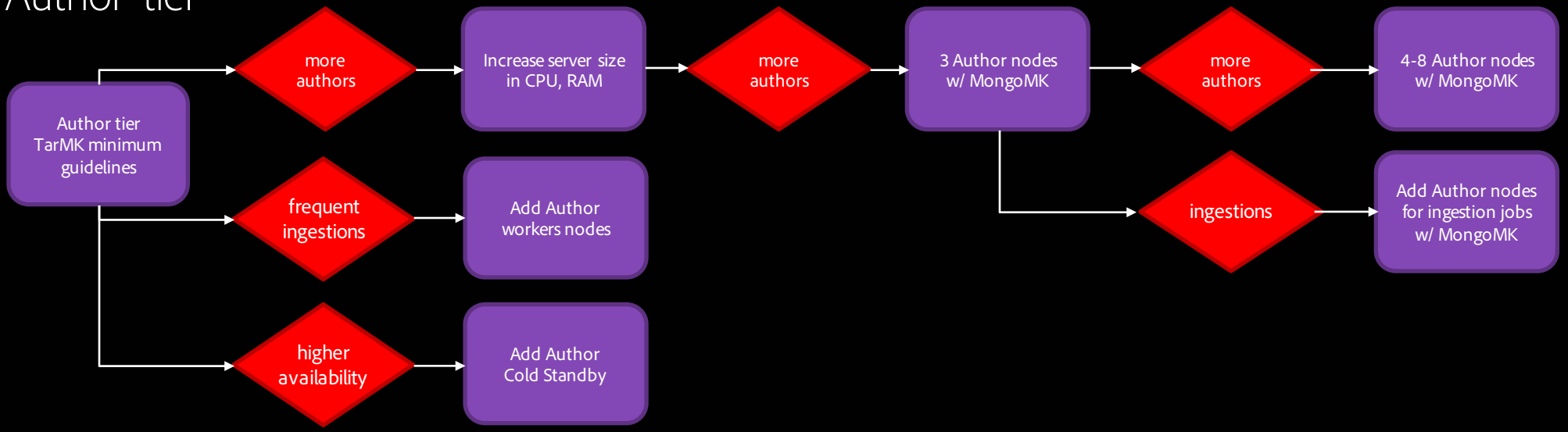
- Purpose-built for content management applications
- Files are always consistent and can be backed up using any file-based backup tool
- Provides a failover mechanism, Cold Standby
- Provides high performance and reliable data storage with minimal operational overhead
- Lower TCO

Criteria for MongoMK

- Number of named users connected in a day: in the thousands or more
- Number of concurrent users: in the hundreds or more
- Volume of asset ingestions per day: in hundreds of thousands or more
- Volume of page edits per day: in hundreds of thousands or more
- Volume of searches per day: in tens of thousands or more

Architecture Scalability Guidelines for AEM Sites + Assets

Author tier



Publish tier

Adobe

Summary of Guidelines for Best Performance

- **TarMK with File Datastore** is the recommended architecture **for most customers**
  - Minimum topology: 1 Author, 2 Publish, 2 Dispatcher
  - Binary-less replication turned on if File Datastore is shared
- **MongoMK with File Datastore** is the recommended architecture **for horizontal scalability of the Author tier**
  - Minimum topology: 3 Author, 3 MongoDB, 2 Publish, 2 Dispatcher
  - Binary-less replication turned on if File Datastore is shared
- **Nodestore** should be stored **on local disk, not NAS**
- **S3** is the recommended datastore **for total volume of assets above 5TB**
  - S3 datastore shared between the Author and Publish tier
  - Binary-less replication turned on
  - Datastore GC requires a first run on all Author & Publish nodes, then a second run on Author
- **Custom index should be created in addition to the OOTB index** based on most common searches
  - Lucene indexes should be used for the custom indexes
- **Customizing workflow can substantially improve the performance**, e.g. Removing video step in the "Update Asset" workflow, disabling listener which are not used, etc.