Stack Allocation	n.a.	*global* < local **Locality** (future) SAFETY: local will not outlive its region NOT ALLOWED: return, escape, global store or capture local parameter isn't requirement, it's behavior guarantee LEGACY NO CAPTURE/NEST: local Applies to non immediate values
	Applies to mutable or mutable nesting data	Applies to functions or function nesting data
Ownership	**Uniqueness** unique < *aliased* (past) LINEAGE: unique has not been aliased ALLOWED: unique may be overwritten	*many* < once **Affinity** (future) SAFETY: once will not alias unique value NOT ALLOWED: call once twice or more LEGACY NO CAPTURE: once, unique
Shared Memory	**Contention** uncontended < shared < *contended* (past) LINEAGE: at most one thread retains uncontended access ALLOWED: • uncontended: thread may read or write • share: thread may only read	*portable* < nonportable **Portability** (future) SAFETY: nonportable will not give access to non *contended* NOT ALLOWED: call nonportable in another thread LEGACY NO CAPTURE: nonportable, uncontended, shared
Effects	n.a.	*unyielding* < yielding **Yielding** (future) SAFETY: yielding function will not perform effect handled in parent stack NOT ALLOWED: yielding ...
Mutable Data	**Visibility** read_write < read < *immutable* (past) LINEAGE: Non *immutable* value ... ALLOWED: • read_write: may be read or written • read: may only be read	*stateless* < observing < stateful **Statefulness** (future) SAFETY: • observing: will not write on read_write data • stateful: will neither read nor write on non *immutable* data NOT ALLOWED: • observing: write • stateful: LEG. NO CAPT.: observing, stateful, read_write, read