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 NO CAPTURE/NEST: local in global 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 NO CAPTURE: once, unique in *many*
Shared Memory	**Contention** *uncontended* < shared < contended (past) LINEAGE: at most one thread retains *uncontended* access ALLOWED / NOT ALLOWED: • *uncontended*: thread may read or write • share: thread may only read • contended, thread may neither read nor write	portable < *nonportable* **Portability** (future) SAFETY: *nonportable* will not give access to non contended NOT ALLOWED: call *nonportable* in another thread NO CAPTURE: *nonportable*, *uncontended*, shared in portable
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