Table 6: Microservice smells

Group	Name (by study)	Category (by study)	DT #1[17] (Arcan)	DT #2[22] (MSANose)	Studies
API Management	API Versioning	MsS		X	[20]
	Static Contract Pitfall	MsP			[20]
	Inadequate Use of APIs	ATD			[8]
	Leak of Service Abstraction	MsP			[20]
Dependency Management	Cyclic Dependency	MsS	X	X	[20]
	Problematic dependency	LoM			[3]
	Inter-service dependency (Ripples)	ChE			[4]
	Microservice coupling	ATD			[8]
	Distributed monolith	AAP			[9]
	Sloth	MsP			[20]
	Integrating Legacy Code	ChE			[4]
	Microservices Integration	ChE			[4]
	Shared Libraries	MsS		X	[20] [6]
	Reusing third-party implementations	ATD			[8]
	Outdated Library	LoM / MTD			[3] [12]
Middleware	ESB usage	MsS		X	[20]
	ESB misuse	ArcS			[15]
	Not Having an API Gateway	MsS / MDT / ArcS		X	[20][12][15]
Discovery	Hard-coded endpoints	MsS / MsP	X	X	[20]
	Manual handling of network issues	ATD			[8]
	Endpoint-based Service Interactions	ArcS			[15]
	Too Many Pont-to-Point (PtP) Connections	MTD			[7] [18]
	Woobly Service Interactions	ArcS			[15]
	Timeout (Dogpiles)	MsP			[20]
Data Management	Shared Persistence	MsS / ArcS	X	X	[20] [15]
	Data ownership	MsP			[20]
	Shared Database	MTD			[12]
	Inappropriate Service Intimacy	MsS		X	[20]
	Insufficient metadata in the messages	ATD			[8]
	Unplanned data sharing and synchronization	ATD			[8]
Decomposition	Microservice Greedy	MsS		X	[20]
	Greedy Service Container	ArcS			[5]
	Multiple Services in One Container	ArcS			[15]
	Grinding Dusty	ArcS			[5]
	Coarse Services	ArcS			[5]
	Wrong Cuts	MsS		X	[20]
	Service Cutting	ChE			[4]
	Envy	MsP			[20]
	Large/complex components	LoM			[3]
	Excessive number of small products	ATD			[8]
	Mega-Service	MsP			[20]
Team/Product Management	Greed	MsP			[20]
	Single Layer Teams	ArcS			[5] [15]
	Adding functionality takes longer	LoM			[3]
	High issue resolution time	LoM			[3]
	Coordination Between Decentralized Teams	ChE			[4]
	Communicating the Importance of Assurance	ChE			[4]

Table 7: Microservice smells (cont.)

Group	**Name** (by study)	**Category** (by study)	**DT #1**[17] (Arcan)	**DT #2**[22] (MSANose)	**Studies**
Architectural Standards	Too Many Standards	MsS		X	[20]
	Lust	MsP			[20]
	Gluttony	MsP			[20]
	Architectural erosion	LoM			[3]
	Duplicate code	LoM			[3]
	No System-Centric View	ChE			[4]
	Mastering Technologies	ChE			[4]
	Technological Heterogeneity	ChE / ATD			[4] [8]
	Architectural/Technical Complexity	ChE / ATD			[4] [8]
	Tool/Process Frustration and Patronization	ChE			[4]
	Distributed Code Repositories	ChE			[4]
	No Standardized Communication Model	MTD / ATD			[18] [7] [8]
	Business Logic Inside Communication Layer	MTD / ATD			[18] [7] [8]
	Different Midlleware Technologies for Communication	MTD			[18] [7]
	Overwhelming amount of unnecessary settings	ATD			[8]
Quality Assurance	Wrath	MsP			[20]
	Pride	MsP			[20]
	Inadequate Testing	LoM / ChE			[3] [4]
	Defects with new releases	LoM			[3]
	Unhealthy Metric Usage	ChE			[4]
DevOps (CI/CD)	Single DevOps Toolchain	ArcS			[5]
	Low release frequency	LoM			[3]
	Inadequate deployment process	LoM			[3]
Documentation	Dismiss Documentation	ArcS			[5]
	Missing / Outdated Documentation	LoM / ChE			[3] [4]
	Weak Source Code and Knowledge management	MTD			[18] [7]
Migration	Thinking Microservices Are a Silver Bullet	MigS			[5]
	Rewrite All Services into Microservices at Once	MigS			[5]
	Learn as You Go	MigS			[5]
	Forgetting About the CAP Theorem	MigS			[5]