INTERNET PAMETINIH UREĐAJA

prof. dr Dejan S. Aleksić

Prirodno-matematički fakultet, Niš

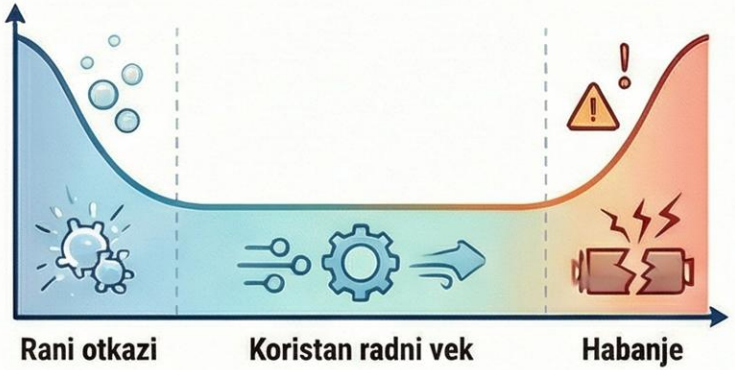
07. ПОУЗДАНОСТ ПОДАТАКА И СИСТЕМА У IOT ОКРУЖЕЊУ

Pouzdanost IoT Sistema: Strategija Odbrane po Dubini

Pouzdanost u IoT okruženjima zahteva sveobuhvatan pristup. Strategija “Odbrane po dubini” primenjuje specifične mehanizme zaštite na svakom sloju arhitekture kako bi se stvorio robustan i otporan sistem.

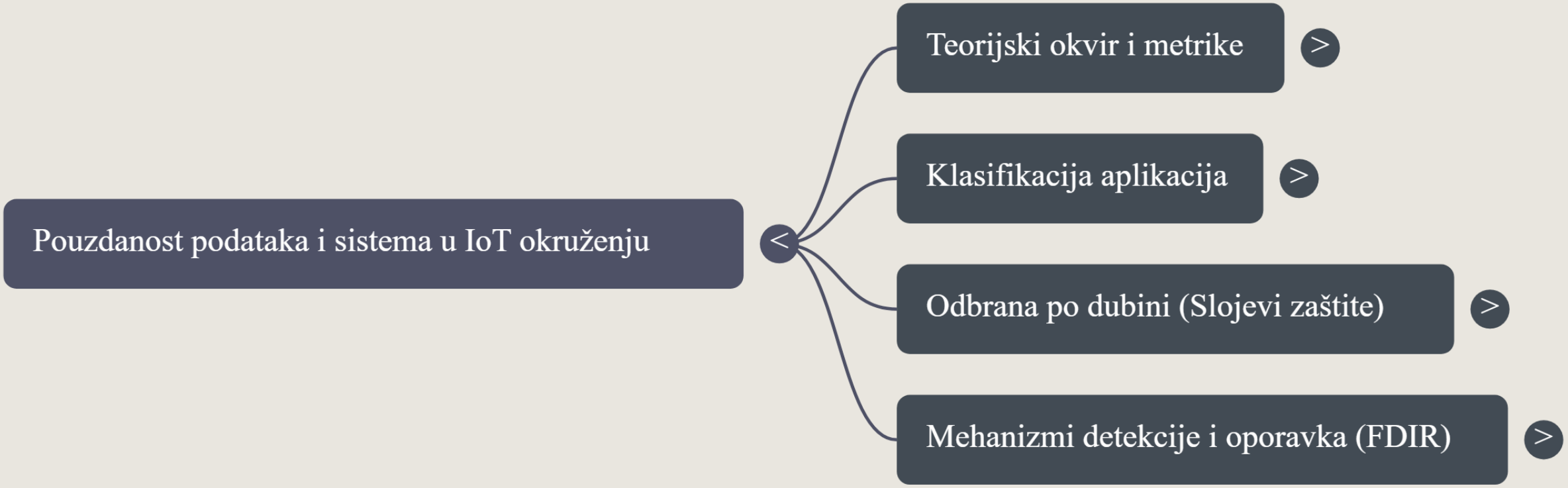
Temelji Pouzdanosti IoT Sistema

Životni Ciklus Otkaza: "Kriva Kade" (Bathtub Curve)



Klasifikacija Sistema po Kritičnosti

Strategija "Odbrane po Dubini"


Pouzdanost podataka i sistema u IoT okruženju
Teorijski okvir i metrike
Klasifikacija aplikacija
Odbrana po dubini (Slojevi zaštite)
Mehanizmi detekcije i oporavka (FDIR)

УВОД У ПОУЗДАНОСТ IOT СИСТЕМА

- Развој Интернета паметних уређаја (Internet of Things – IoT) редефинисао је парадигму интеракције информационих система са физичким окружењем.
- Уместо пасивног бележења података, савремени IoT системи врше аквизицију, обраду и актуацију у реалном времену, постајући тиме интегрални део критичне инфраструктуре у индустрији, здравству и енергетици.
- Кључно је уочити суштинску разлику између традиционалног веб развоја и IoT инжењерства.
- Реч је о дистрибуираним системима сачињеним од великог броја хетерогених уређаја лимитираних ресурса, који комуницирају путем нестабилних бежичних канала.
- У таквом контексту, откази се не посматрају као изузеци (аномалије), већ као очекивано, номинално стање система.
- Циљ се, стога, помера са покушаја апсолутне превенције отказа на пројектовање система отпорних на грешке (fault-tolerant), који могу одржати функционалност и у деградираним условима.

УВОД У ПОУЗДАНОСТ IOT СИСТЕМА

- Поузданост у IoT-у је вишеслојан, свеобухватан проблем који се не може редуковати искључиво на стабилност мрежне конекције.
- Он обухвата целокупан технолошки стек:
  - од физичке деградације сензора услед фактора околине,
  - преко стабилности фирмвера у реалном времену,
  - до конзистентности података у дистрибуираним Cloud системима.
- С обзиром на то да ови уређаји често оперишу аутономно, без могућности физичког приступа сервисера, имплементација механизама за аутоматску детекцију и опоравак од грешака (FDIR) постаје императив.

ДЕФИНИЦИЈА ПОУЗДАНОСТИ И ДОСТУПНОСТИ

- У теорији система, фундаментална мера је поузданост R(t), коју формално дефинишемо као вероватноћу да систем исправно функционише у континуитету током временског интервала [0, t], под претпоставком да је у тренутку t = 0 био исправан.
- Међутим, у пракси, а нарочито у контексту сервисно оријентисаних IoT архитектура, метрика поузданости често није довољна.
- Корисника сервиса примарно занима доступност (Availability), односно вероватноћа да ће систем бити оперативан у тренутку када му је потребан. Доступност A се изводи из два кључна параметра времена:
  - MTTF (Mean Time To Failure): Просечно време које протеке пре него што компонента откаже. Ово је директна мера квалитета и дуготрајности уграђеног хардвера.
  - MTTR (Mean Time To Recover): Просечно време потребно да се настали квар детектује, изолује и систем врати у оперативно стање (било аутоматским ресетом или заменом компоненте).

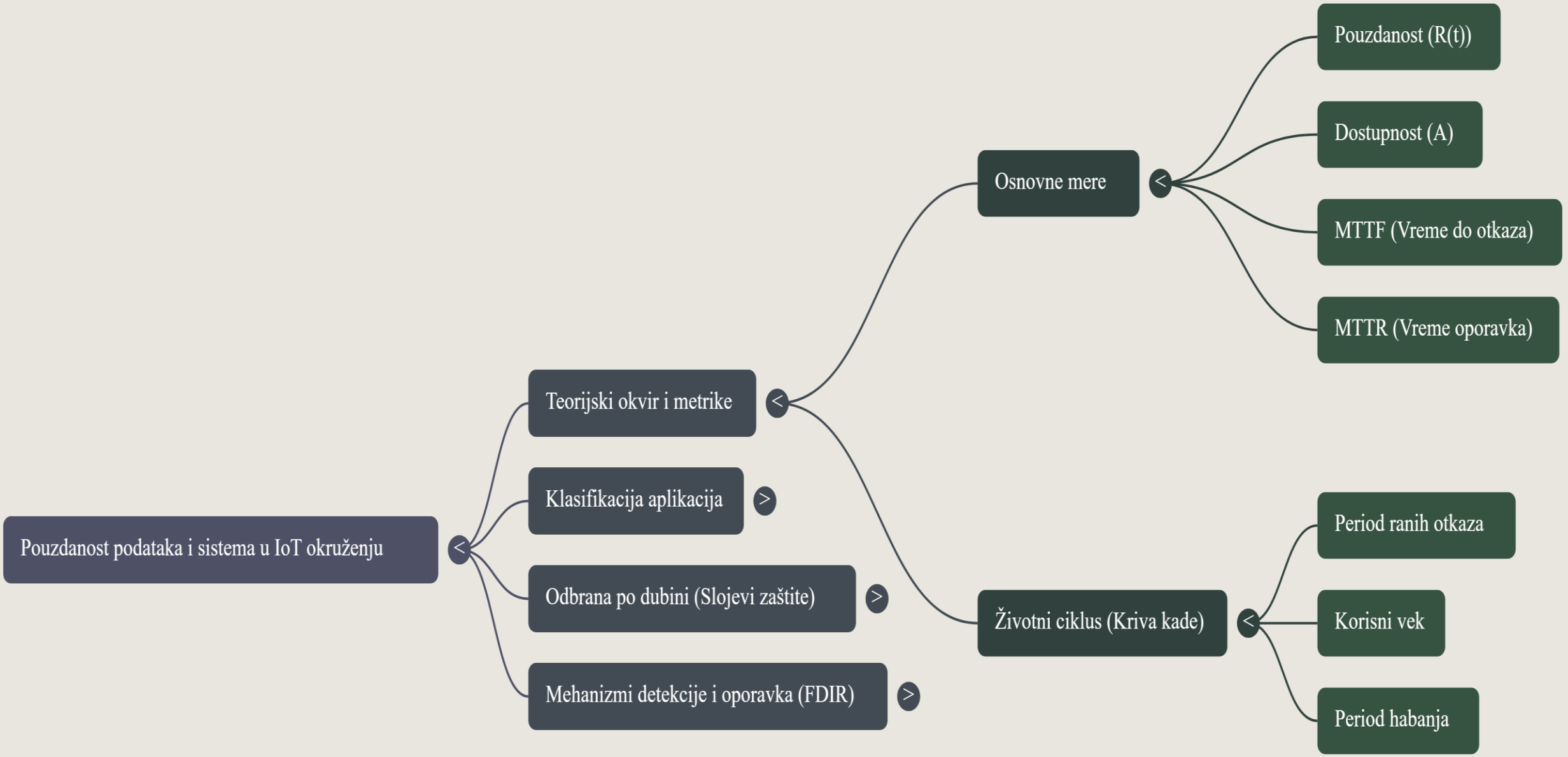
ДЕФИНИЦИЈА ПОУЗДАНОСТИ И ДОСТУПНОСТИ

- Веза између ових величина дефинисана је формулом која идентификује два кључна параметра путем којих је могуће управљати доступношћу система:

$$A = \frac{MTTF}{MTTF + MTTR}$$

- Из једначине следи важан закључак: високу доступност можемо постићи или улагањем у скуп, ултра-поуздан хардвер (повећање MTTF) или, што је чешћи случај у IoT свету, имплементацијом механизама за брз и аутоматски опоравак (смањење MTTR).

УВОД У ПОУЗДАНОСТ IOT СИСТЕМА



КЛАСИФИКАЦИЈА АПЛИКАЦИЈА ПРЕМА КРИТИЧНОСТИ

- Системи нулте толеранције (Safety-Critical)

  За овакве системе очекивано време до отказа мора бити дуже од предвиђеног животног века уређаја (MT TF > Tmission). Овде се не можемо ослонити на механизме опоравка, јер је сваки прекид рада фаталан.

  Приступ: Једино решење је масивна хардверска редунданса. Систем мора поседовати својство FailOperational — способност да настави исправан рад чак и када једна његова компонента потпуно откаже.
- Рестартабилни системи (Mission-Critical / High Availability)

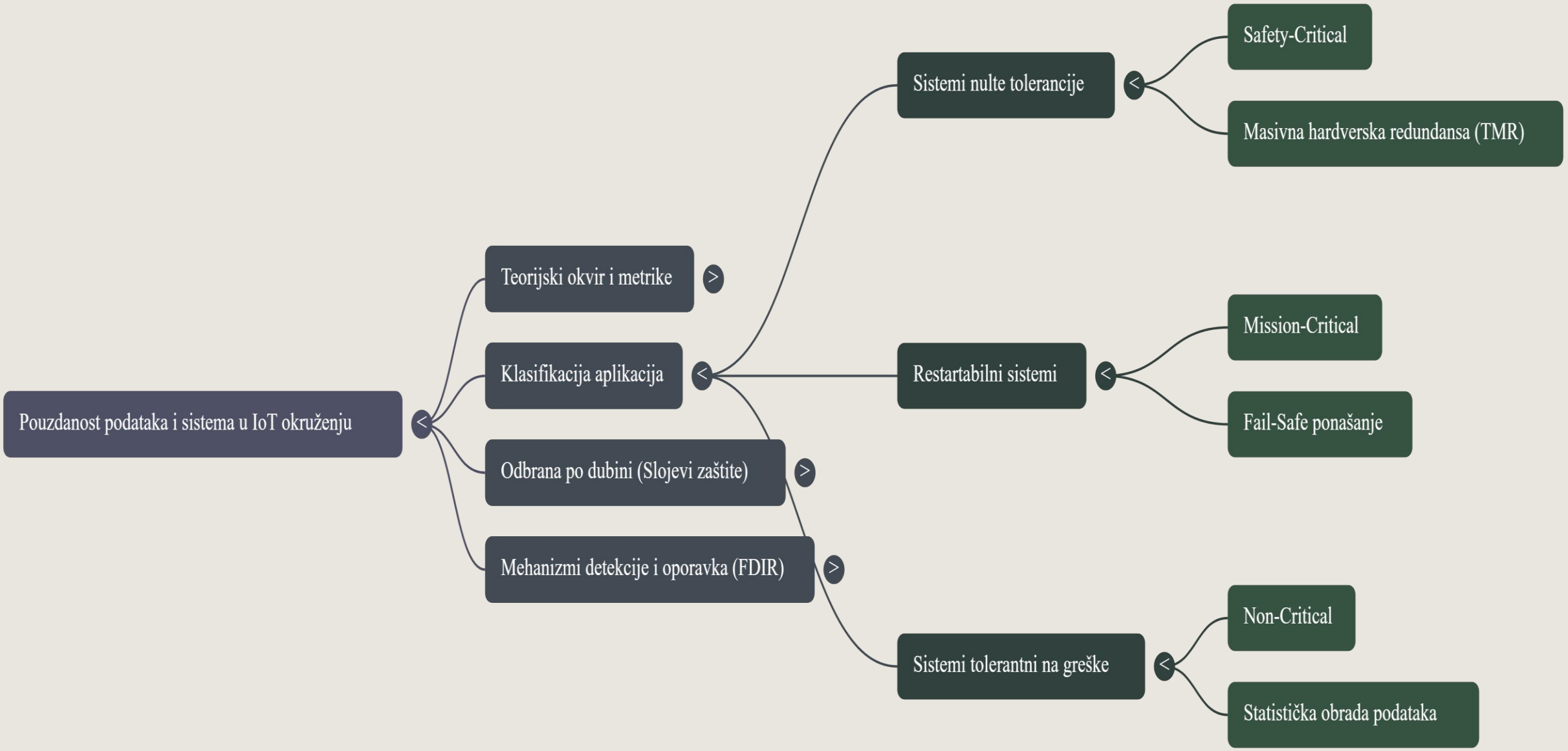
  овде отказ не угрожава безбедност, већ доводи до значајних финансијских губитака или прекида кључних пословних процеса.

  Приступ: Инжењерски фокус се помера са спречавања квара (MTTF) на брзину опоравка (минимизација MTTR-a).
- Системи толерантни на грешке (Non-Critical)

  Ова категорија обухвата масивне IoT примене где вредност није у појединачном податку, већ у статистичком скупу.

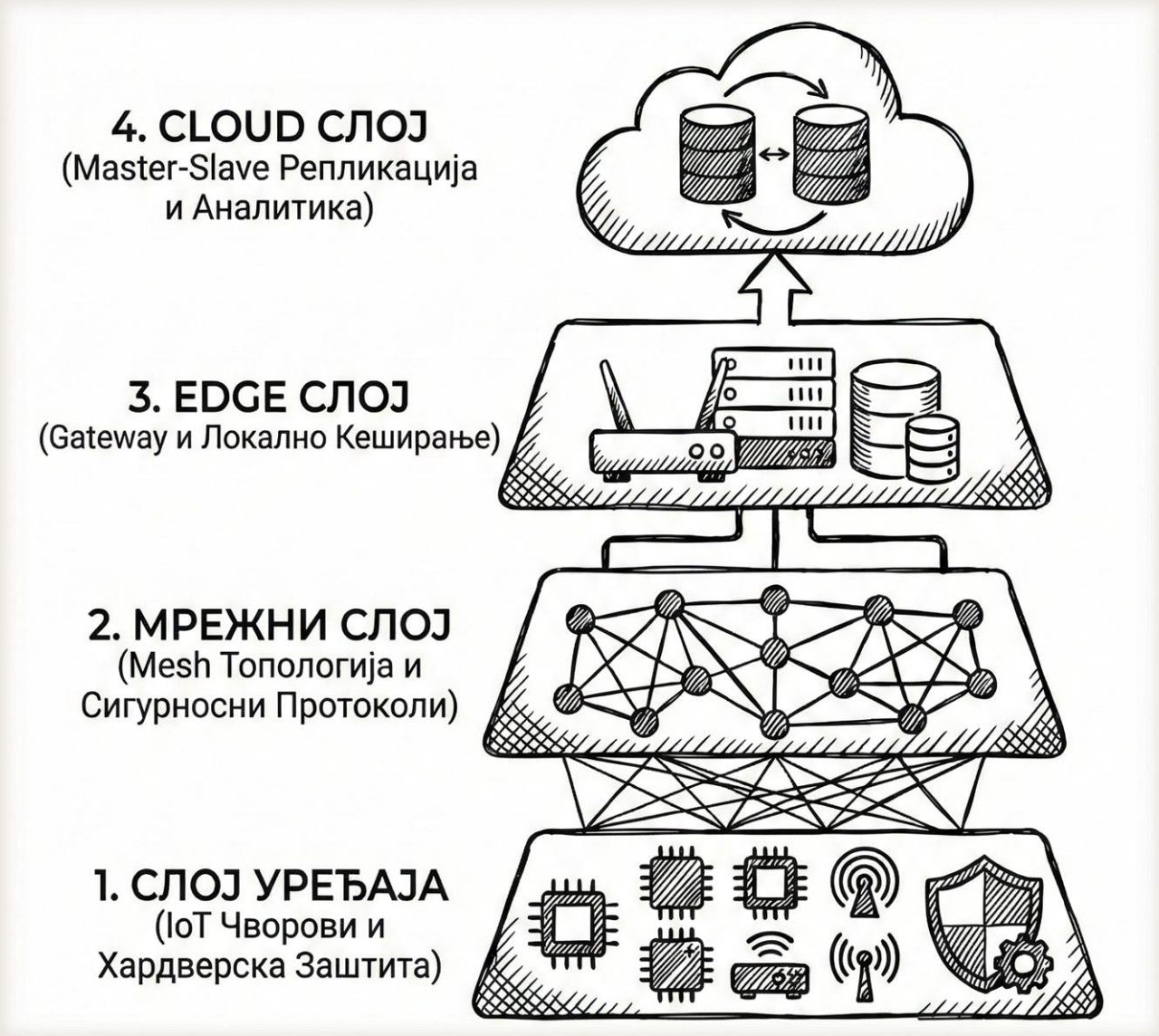
  Приступ: Улагање у скуп хардвер или редундансу овде је бацање ресурса. Поузданост се постиже на серверској страни, коришћењем алгоритама за детекцију аномалија и статистичку обраду.

УВОД У ПОУЗДАНОСТ IOT СИСТЕМА



СТРАТЕГИЈА „ОДБРАНЕ ПО ДУБИНИ”

- Како бисмо изградили систем који је заиста отпоран на грешке, не смемо се ослонити на једну тачку отпорности (Single Point of Failure).
- Уместо тога, примењујемо принцип „одбране по дубини” (Defense in Depth).
- Овај концепт у контексту поузданости налаже да сваки слој IoT архитектуре мора поседовати сопствене, аутономне механизме за детекцију и митигацију отказа.
- Систем мора бити пројектован тако да пробој или отказ једне баријере (нпр. прекид Wi-Fi везе) не доведе до колапса целог система, већ да га „амортизује” наредни слој заштите (нпр. локално кеширање на Edge гејтвеју).
- Не веруј слоју изнад, нити слоју испод.



УВОД У ПОУЗДАНОСТ IOT СИСТЕМА

ФИЗИЧКИ НИВО И НИВО УРЕЂАЈА (DEVICE LAYER)

- Трајни откази (Hard Failures) - попут физичког прегоревања компоненте, где је једини лек замена уређаја.

- Пролазни откази (Soft Failures) – Уређај је физички исправан, али је ушао у недефинисано стање рада и више не извршава инструкције коректно.

ФИЗИЧКИ НИВО И НИВО УРЕЂАЈА (DEVICE LAYER)

- Watchdog Тајмери (WDT)
- Детекција пада напона (Brownout Detection)
- Заштита Flash меморије (Wear Leveling)
- Хардверска редунданса (Hot Standby)
- Дијагностика и валидација паметних сензора
  - Верификација идентитета уређаја (WHO_AM_I Check)
  - Провера интегритета података (CRC/Checksum)
  - Мониторинг статусних регистар
  - Сензорска редунданса и унакрсна валидација (Cross-Validation)

УВОД У ПОУЗДАНОСТ IOT СИСТЕМА

МРЕЖНИ НИВО И КОМУНИКАЦИЈА (NETWORK LAYER)

- Комуникациони подсистем представља најнестабилнију компоненту IoT стека.
- За разлику од детерминистичких, жичаних веза у индустријским постројењима, IoT уређаји најчешће оперишу у оквиру онога што се дефинише као LLN (Low-Power and Lossy Networks).
- То су мреже које карактеришу висока стопа губитка пакета, нестабилне везе и строга енергетска ограничења.
- Поузданост на овом нивоу није гарантована физичким медијумом, већ се мора синтетизовати кроз интелигентне топологије и робусне комуникационе протоколе.

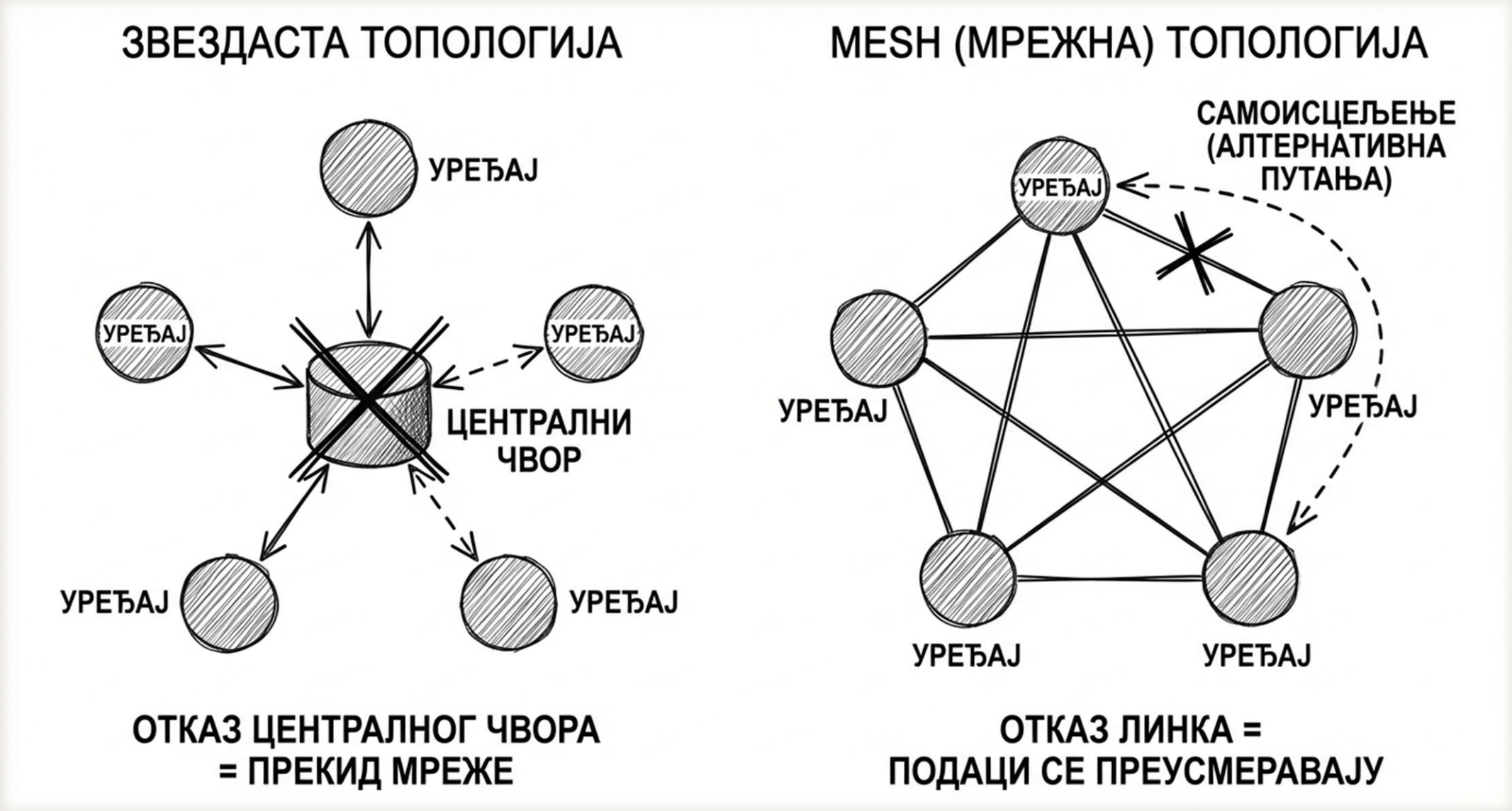
ТОПОЛОГИЈЕ МРЕЖЕ И ОТПОРНОСТ НА ОТКАЗ

- Архитектура повезивања чворова директно диктира способност система да преживи отказ појединачних уређаја или радио-сметње.
- Разликујемо два доминантна приступа:

  - Звездаста топологија (Star Topology) - Ово је централизована архитектура (карактеристична за Wi-Fi и LoRaWAN) где сви крајњи уређаји комуницирају директно са централним гејтвејем.

  - Mesh топологија (Self-Healing Networks) - У Mesh мрежама (као што су ZigBee или Thread), сваки чвор са сталним напајањем има двоструку улогу: он је и извор података и рутер који прослеђује саобраћај за друге чворове.

ТОПОЛОГИЈЕ МРЕЖЕ И ОТПОРНОСТ НА ОТКАЗ



ПРОТОКОЛИ И QOS (QUALITY OF SERVICE)

- Транспортни слој: UDP vs. TCP

  Иако TCP (Transmission Control Protocol) нуди гарантовану испоруку и уређен редослед пакета, он је често „претежак” за IoT због великог заглавља и начина успостављања везе (Handshake). С друге стране, UDP је бржи али непоуздан. Савремени IoT протоколи решавају ову дилему имплементацијом поузданости на апликативном слоју.

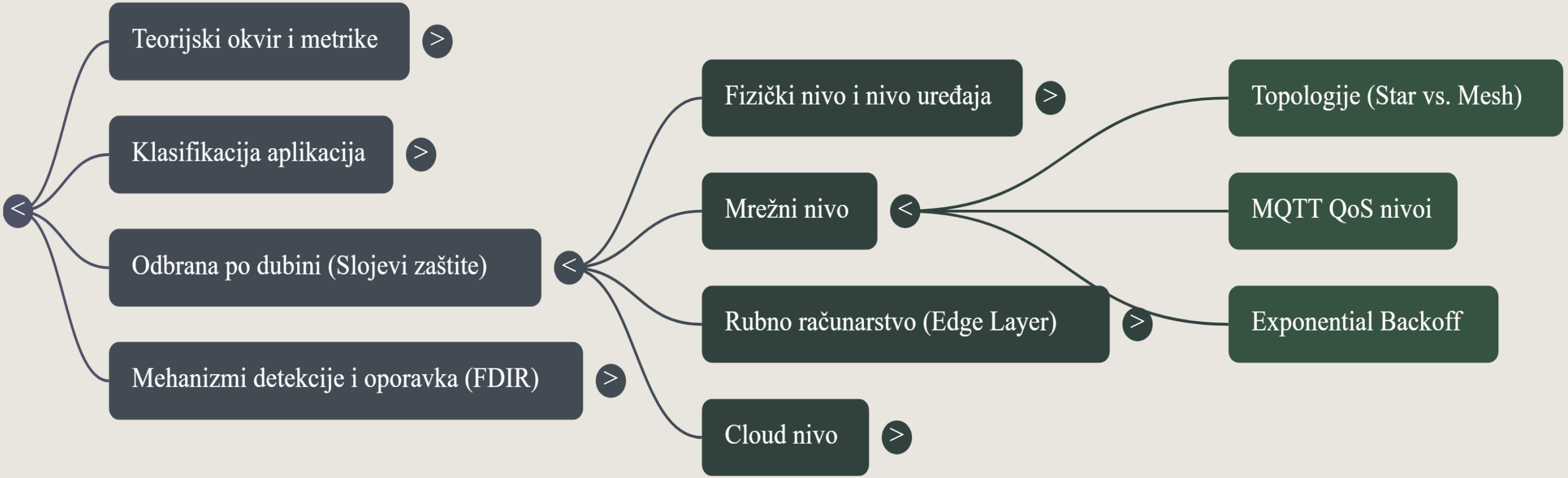
ПРОТОКОЛИ И QOS (QUALITY OF SERVICE)

- MQTT QoS Нивои –
  - **QoS 0 (At most once):** „Fire and forget”. Порука се шаље једном, без потврде пријема. Користи се за некритичне сензорске податке где је повремени губитак узорка прихватљив ради уштеде енергије.
  - **QoS 1 (At least once):** Гарантује да ће порука стићи, али дозвољава дупликате. Пошиљалац чува поруку све док не добије PUBACK потврду. Ово је најчешћи избор јер нуди добар баланс поузданости и ефикасности, уз услов да апликација мора бити идемпотентна.
  - **QoS 2 (Exactly once):** Највиши ниво поузданости који гарантује тачно једну испоруку кроз четворостепену размену пакета (4-way handshake). Због високе цене у мрежном саобраћају и латенцији, користи се само за критичне команде (нпр. трансакције плаћања или искључивање индустријске машине).

СТРАТЕГИЈЕ РЕТРАНСМИСИЈЕ И КОНТРОЛА ЗАГУШЕЊА

- Није довољно само детектовати губитак пакета; начин на који систем покушава поновно слање (retry mechanism) може бити пресудан за стабилност целе мреже.
- Наиван приступ, где уређај одмах и константно понавља слање након грешке, често доводи до ефекта „мрежне олује” (Broadcast Storm).
- Ако стотине уређаја истовремено изгубе везу са сервером (нпр. Након рестарта гејтвеја) и сви истовремено покушају реконекцију, мрежа ће доживети колапс услед загушења.

УВОД У ПОУЗДАНОСТ IOT СИСТЕМА



РУБНО РАЧУНАРСТВО (EDGE/FOG LAYER)

- Традиционални „Cloud-centric” приступ, у којем „глупи” сензори шаљу сирове податке директно на удаљени сервер, показао се као недовољно робустан за критичне примене.
- Ослањање искључиво на доступност WAN (Wide Area Network) везе уводи значајан ризик: прекид интернет конекције значи губитак функционалности и потенцијални губитак података.
- Како би се овај ризик митигирао, модерне архитектуре уводе слој Рубног рачунарства (Edge Computing).
- Овај слој, најчешће реализован кроз паметне мрежне пролазе или индустријске рачунаре локацијски блиске извору података, преузима улогу локалног контролера и привременог складишта података.
- Са аспекта поузданости, Edge слој обезбеђује два кључна механизма:
    - аутономију рада и
    - асинхрону синхронизацију

РУБНО РАЧУНАРСТВО (EDGE/FOG LAYER)

- Локална аутономија (Local Operation Fallback)

  Основни принцип пројектовања поузданог IoT система гласи:

  контролна петља мора бити затворена што ближе извору.

  Уколико систем управља критичним процесом (нпр. активирање система за гашење пожара или заустављање производне траке), логика одлучивања не сме зависити од удаљеног Cloud сервера.

  Edge уређај мора поседовати копију конфигурације и пословне логике („Digital Twin” стање) која му омогућава да доноси валидне одлуке чак и када је веза са интернетом у потпуности прекинута.

  Cloud платформа се у овом моделу користи искључиво за дугорочну аналитику, извештавање и глобалну оркестрацију, док је оперативна контрола делегирана на руб мреже

РУБНО РАЧУНАРСТВО (EDGE/FOG LAYER)

- Механизам „Чувај и проследи” (Store and Forward)

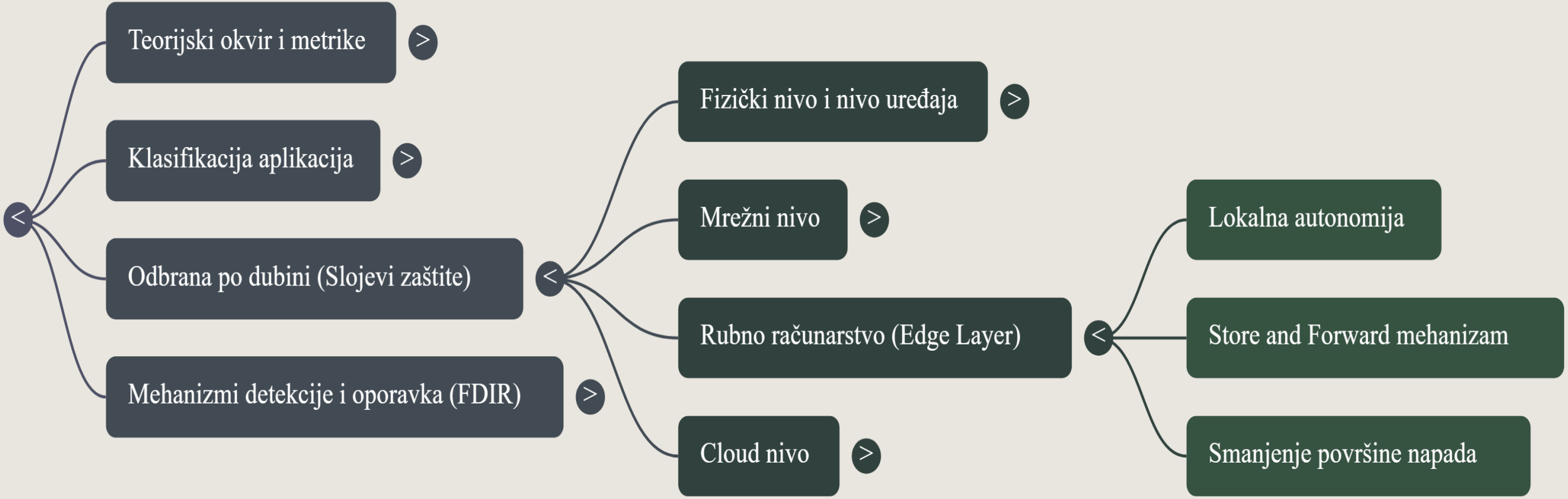
  Губитак мрежне везе не сме резултирати губитком прикупљених података. Рубни мрежни пролази (Edge Gateways) имплементирају Store and Forward механизам који функционише као привремени бафер између нестабилне локалне мреже и Cloud-а.

  1. Детекција прекида: Када мрежни пролаз детектује немогућност испоруке пакета (након исцрпљивања покушаја ретрансмисије), он прелази у Offline режим.
  2. Локално кеширање: Подаци се уместо слања преусмеравају у локалну перзистентну меморију (нпр. SQLite базу или кружни бафер на Flash меморији).
  3. Очување временског контекста: Критично је важно да се уз сваки податак сачува и оригинални временски жиг (Timestamp) тренутка настанка догађаја, а не тренутка слања. Без овога, накнадна анализа историјских података била би бескорисна.
  4. Синхронизација: Када се веза поново успостави, мрежни пролаз започиње процес пражњења бафера, шаљући акумулиране податке Cloud платформи, често користећи механизме компресије ради уштеде пропусног опсега

РУБНО РАЧУНАРСТВО (EDGE/FOG LAYER)

- Сажимање података и смањење „површине напада”
- Поред заштите од губитка везе, Edge слој игра критичну улогу у безбедносном аспекту поузданости кроз минимизацију такозване „површине напада” (Attack Surface).
- Овај појам означава укупан збир свих тачака (отворених портова, активних конекција, API крајњих тачака) путем којих неовлашћени актер може покушати да компромитује систем. У класичној архитектури где се стотине сензора директно повезују на интернет, сваки од њих представља потенцијалну улазну тачку за напад (нпр. DDoS напад који обара сервис).
- Увођењем рубног мрежног пролаза (Edge Gateway), имплементира сеархитектура „левка”: велики број сензорских чворова, који су често безбедносно рањиви, изолује се иза једног робусног уређаја који врши агрегацију података. Уместо успостављања стотина отворених конекција ка Cloud-у, одржава се само једна — безбедна и шифрована веза.

РУБНО РАЧУНАРСТВО (EDGE/FOG LAYER)



CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА

- На највишем слоју IoT архитектуре, фокус инжењера помера се са проблема конективности на проблеме интегритета, доступности и складиштених података.
- Иако савремене Cloud платформе нуде апстракцију инфраструктуре, погрешно је претпоставити да је поузданост у облаку загарантована „по аутоматизму”.
- Напротив, робусност апликације на овом нивоу зависи искључиво од архитектуре коју пројектант имплементира.
- Три кључна механизма за обезбеђивање поузданости на серверској страни су:
  1. репликација података,
  2. асинхроно процесирање путем редова чекања и
  3. управљање конзистентношћу.

CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА

1. Репликација и дистрибуција података

У пракси се најчешће примењује архитектура Master-Slave репликације (или у новијој номенклатури Primary-Replica).

У овом моделу:

- **Master (Примарна база):** Једина инстанца која прихвата операције уписа (Write). Ово осигурава да не дође до конфликата података.
- **Slave (Реплика):** Једна или више инстанци које служе само за читање (Read-only) и које асинхроно преузимају промене са Мастера.

- Са аспекта поузданости, ова архитектура нуди механизам за преусмеравање на резервни систем (Failover).
- Уколико примарна база доживи хардверски отказ, аутоматизовани надзорни систем промовише једну од реплика у новог Мастера.
- Иако може доћи до микронског губитка података који још нису стигли да се реплицирају, систем наставља са радом, чиме се обезбеђује континуитет сервиса.

CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА

2. Редови порука и асинхрона обрада (Message Queuing)

- IoT системи су подложни наглим, стохастичким скоковима у саобраћај (Traffic Spikes).
- Директно, синхроно уписивање ових података у базу довело би до тренутног загушења и потенцијалног пада система.
- Ради повећања поузданости, у архитектуру се уводи компонента за асинхроно раздвајање (Decoupling) — ред порука (нпр. Apache Kafka или RabbitMQ).
- Овај механизам функционише по принципу прихватног бафера:
  1. Пристигли подаци се тренутно смештају у ред порука, који је оптимизован за прихват великог броја захтева у кратком року.
  2. Потрошачи (Consumers) — сервиси који обрађују податке — преузимају поруке из реда сопственим темпом, онолико брзо колико могу да их обраде.

- У случају успорења базе података, ред порука се привремено пуни, али подаци нису изгубљени и систем остаје оперативан.
- Овај образац, познат као „нивелисање оптерећења” (Load Leveling), кључан је за стабилност система под оптерећењем.

CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА

- Микросервисна архитектура и изолација отказа
  - Савремене IoT платформе у облаку ретко се имплементирају као монолитни системи. Уместо тога, доминантна је архитектура микросервиса (Microservices Architecture), где је функционалност система декомпонована на скуп независних сервиса (нпр. сервис за аутентификацију, сервис за временске серије, сервис за аларме).
  - Иако ова архитектура повећава скалабилност, она уводи ризик од каскадних отказа (Cascading Failures).
  - Уколико један некритичан сервис (нпр. модул за генерисање извештаја) постане недоступан или спор, он може исцрпети дељене ресурсе (попут нити процесора или конекција ка бази) и тиме узроковати прекид рада критичних сервиса (нпр. пријем података о пожару).

  - Образац заштитних преграда (Bulkhead Pattern)
  - Постепена деградација (Graceful Degradation)
  - Оркестрација и верификација исправности (Liveness & Readiness Probes)

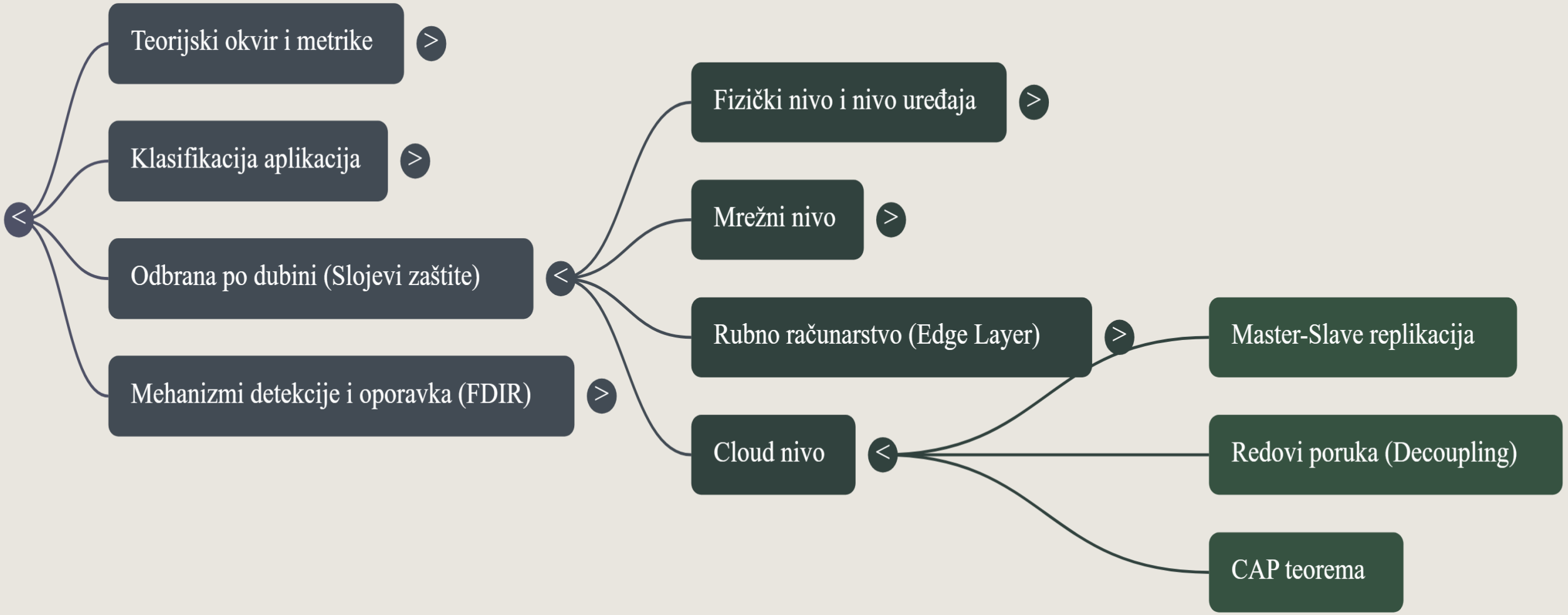
CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА

- Микросервисна архитектура и изолација отказа
  - **Образац заштитних преграда (Bulkhead Pattern)** - исцрпљивање ресурса у једном микросервису не сме довести до колапса целокупне платформе
  - **Постепена деградација (Graceful Degradation)** - приликом отказа зависног сервиса не дође до потпуног прекида рада, већ до преласка у режим рада са умањеним функционалностима
  - **Оркестрација и верификација исправности (Liveness & Readiness Probes)** - Обезбеђивање поузданости микросервиса данас се аутоматизује коришћење платформи за оркестрацију контејнера (као што је Kubernetes).

CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА

- Микросервисна архитектура и изолација отказа
  - **Оркестрација и верификација исправности (Liveness & Readiness Probes)** - Обезбеђивање поузданости микросервиса данас се аутоматизује коришћење платформи за оркестрацију контејнера (као што је Kubernetes). Уместо пасивног рада, ове платформе активно надзиру стање сервиса применом два дистинктна механизма верификације:
    - **Провера виталности (Liveness Probe)** - Периодични дијагностички упит којим платформа утврђује да ли је процес извршавања сервиса активан. Уколико сервис уђе у стање блокаде (Deadlock) и престане да одговара на ове упите, оркестратор детектује нефункционалност и аутоматски покреће процедуру поновног покретања (Restart) како би опоравио сервис.
    - **Провера оперативности (Readiness Probe)** - Механизам којим се утврђује да ли је сервис спреман да прихвати и обради кориснички саобраћај. Сервис може бити активан (виталан), али тренутно преоптерећен или у фази иницијализације (нпр. учитавање великих модела у меморију). У таквом сценарију, платформа га привремено искључује из балансера оптерећења (Load Balancer), чиме се спречава неуспешна обрада захтева све док се сервис не стабилизује.

CLOUD НИВО И УПРАВЉАЊЕ ПОДАЦИМА



МЕХАНИЗМИ ДЕТЕКЦИЈЕ И ОПОРАВКА (FDIR)

- У аутономним системима, пасивно чекање да оператер примети квар није прихватљива стратегија. Инжењерски одговор на овај изазов је имплементација FDIR петље (Fault Detection, Isolation, and Recovery), која представља својеврсни „имуни систем” IoT архитектуре.
- Овај концепт подразумева три корака:
  1. правовремену детекцију аномалије,
  2. изолацију неисправне компоненте како би се спречила пропагација грешке,
  3. аутоматски опоравак (најчешће путем рестарта или преласка на резервни систем).

МЕХАНИЗМИ ДЕТЕКЦИЈЕ И ОПОРАВКА (FDIR)

- Механизам периодичне потврде активности (Heartbeat) –
  - У дистрибуираним системима, најтеже је детектовати тзв. „тихи отказ” — ситуацију када сензор или сервис једноставно престане да шаље податке, без слања експлицитне поруке о грешци.
  - Да бисмо разликовали „тишину” услед недостатка догађаја од „тишине” услед квара, користимо механизам периодичне потврде активности (Heartbeat).
  - Принцип рада заснива се на детерминистичком слању статусних порука („пулсева”) у фиксним временским интервалима.
  - Пријемна страна (нпр. сервер) конфигурише тајмер за детекцију отказа.
  - Критични инжењерски параметар је однос између ова два времена. Пракса налаже да праг толеранције мора бити већи од периода слања, како би се избегли лажни позитиви узроковани кашњењем мреже (тзв. Jitter).

МЕХАНИЗМИ ДЕТЕКЦИЈЕ И ОПОРАВКА (FDIR)

- Идемпотентност операција –
  - Због непоузданости транспортног слоја, IoT протоколи (попут MQTT QoS 1) често користе стратегију поновног слања порука (Retry Mechanism).
  - Ово неминовно доводи до ситуације где сервер прима дупликате исте поруке.
  - Ако операција није пројектована коректно, обрада дупликата може корумпирати стање система.
  - Овде уводимо математички концепт идемпотентности.
  - Операција се сматра идемпотентном ако њено вишеструко извршавање над истим скупом података производи идентичан резултат као и једноструко извршавање.
  - Све критичне команде у IoT системима морају бити дизајниране као идемпотентне, често коришћењем јединствених идентификатора захтева (Request ID) за дедупликацију

МЕХАНИЗМИ ДЕТЕКЦИЈЕ И ОПОРАВКА (FDIR)

- Образац заштитног прекидача (Circuit Breaker) –
  - Када удаљени сервис (нпр. база података) постане недоступан или преоптерећен, најгоре што клијентски уређаји могу да ураде јесте да наставе са агресивним покушајима поновног повезивања.
  - Ово доводи до ефекта „лавине” који спречава опоравак сервиса.
  - Да бисмо ово спречили, имплементирамо софтверски образац заштитног прекидача (Circuit Breaker Pattern).
  - Овај механизам функционише аналогно аутоматском осигурачу у електротехници:
    1. **Затворено (Closed):** Саобраћај тече нормално. Систем броји грешке.
    2. **Отворено (Open):** Када број грешака пређе дефинисани праг (нпр. 5 грешака у 10 секунди), прекидач се „отвара”. Сви наредни покушаји комуникације се тренутно блокирају без слања захтева мрежом, чиме се штеде ресурси.
    3. **Полу-отворено (Half-Open):** Након периода хлађења, систем пропушта само један пробни захтев. Ако он успе, прекидач се поново затвара; ако не успе, враћа се у отворено стање.
  - Овај механизам омогућава преоптерећеном систему неопходно време за самоисцељење

МЕХАНИЗМИ ДЕТЕКЦИЈЕ И ОПОРАВКА (FDIR)